W3School Bootstrap 教程

wizardforcel

Published with GitBook



目錄

W3School Bootstrap 教程

来源：Bootstrap 教程

整理：飞龙

Bootstrap 简介

Bootstrap 简介

什么是 Bootstrap？

Bootstrap 是一个用于快速开发 Web 应用程序和网站的前端框架。Bootstrap 是基于 HTML、CSS、JAVASCRIPT 的。

历史

Bootstrap 是由 *Twitter* 的 *Mark Otto* 和 *Jacob Thornton* 开发的。Bootstrap 是 2011 年八月在 GitHub 上发布的开源产品。

为什么使用 Bootstrap？

- 移动设备优先：自 Bootstrap 3 起，框架包含了贯穿于整个库的移动设备优先的样式。
- 浏览器支持：所有的主流浏览器都支持 Bootstrap。



- 容易上手：只要您具备 HTML 和 CSS 的基础知识，您就可以开始学习 Bootstrap。
- 响应式设计：Bootstrap 的响应式 CSS 能够自适应于台式机、平板电脑和手机。更多有关响应式设计的内容详见 Bootstrap 响应式设计。



- 它为开发人员创建接口提供了一个简洁统一的解决方案。
- 它包含了功能强大的内置组件，易于定制。
- 它还提供了基于 Web 的定制。
- 它是开源的。

Bootstrap 包的内容

- 基本结构：Bootstrap 提供了一个带有网格系统、链接样式、背景的基本结构。这将在 **Bootstrap** 基本结构 部分详细讲解。
- **CSS**：Bootstrap 自带以下特性：全局的 CSS 设置、定义基本的 HTML 元素样式、可扩展的 class，以及一个先进的网格系统。这将在 **Bootstrap CSS** 部分详细讲解。
- 组件：Bootstrap 包含了十几个可重用的组件，用于创建图像、下拉菜单、导航、警告框、弹出框等等。这将在 布局组件 部分详细讲解。
- **JavaScript** 插件：Bootstrap 包含了十几个自定义的 jQuery 插件。您可以直接包含所有的插件，也可以逐个包含这些插件。这将在 **Bootstrap** 插件 部分详细讲解。
- 定制：您可以定制 Bootstrap 的组件、LESS 变量和 jQuery 插件来得到您自己的版本。

在线实例

本站的 Bootstrap 教程包含了上百个实例。

你可以使用我们的在线编辑器在线编辑代码，并点击运行按钮查看结果。

Bootstrap 实例

```
<div class="container">
  <p>Create a responsive table with alternating cell background col

  <div class="table-responsive">
    <table class="table table-striped table-bordered">
      <thead>
        <tr>
          <th>#</th>
          <th>Name</th>
          <th>Street</th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td>1</td>
          <td>Anna Awesome</td>
          <td>Broome Street</td>
        </tr>
        <tr>
          <td>2</td>
          <td>Debbie Dallas</td>
          <td>Houston Street</td>
        </tr>
        <tr>
          <td>3</td>
          <td>John Doe</td>
          <td>Madison Street</td>
        </tr>
      </tbody>
    </table>
  </div>

</div>
```

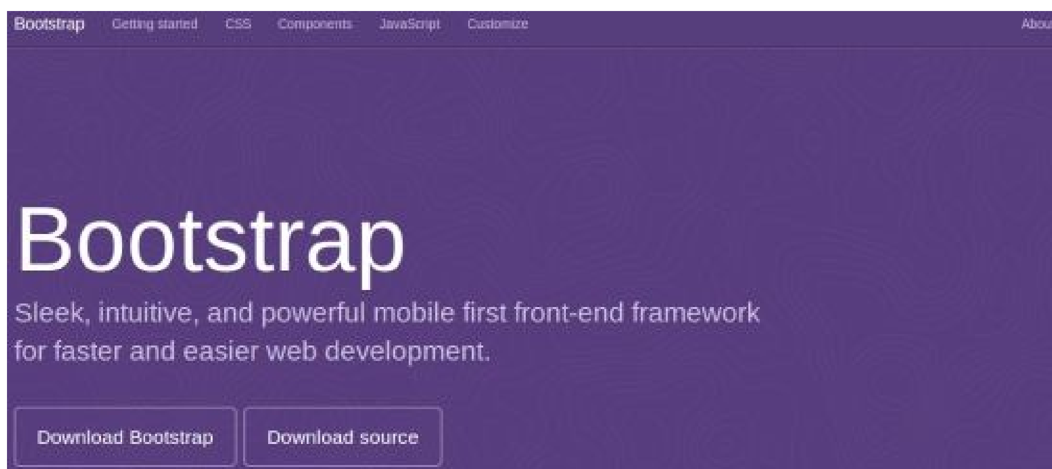
点击 "尝试一下" 按钮查看它是如何工作的。

Bootstrap 环境安装

Bootstrap 安装是非常容易的。本章将讲解如何下载并安装 Bootstrap，讨论 Bootstrap 文件结构，并通过一个实例演示它的用法。

下载 Bootstrap

您可以从 http://getbootstrap.com/ 上下载 Bootstrap 的最新版本。当您点击这个链接时，您将看到如下所示的网页：



您会看到两个按钮：

- *Download Bootstrap*：下载 Bootstrap。点击该按钮，您可以下载 Bootstrap CSS、JavaScript 和字体的预编译的压缩版本。不包含文档和最初的源代码文件。
- *Download Source*：下载源代码。点击该按钮，您可以直接从 from 上得到最新的 Bootstrap LESS 和 JavaScript 源代码。

如果您使用的是未编译的源代码，您需要编译 LESS 文件来生成可重用的 CSS 文件。对于编译 LESS 文件，Bootstrap 官方只支持 Recess，这是 Twitter 的基于 less.js 的 CSS 提示。

为了更好的了解和更方便的使用，我们将在本教程中使用 Bootstrap 的预编译版本。

由于文件是被编译过和压缩过的，在独立的功能开发中，您不必每次都包含这些独立的文件。

本教程编写时，使用的是最新版（Bootstrap 3）。

文件结构

预编译的 Bootstrap

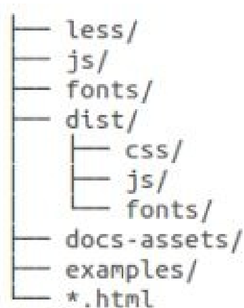
当您下载了 Bootstrap 的已编译的版本，解压缩 ZIP 文件，您将看到下面的文件/目录结构：



如上图所示，可以看到已编译的 CSS 和 JS（bootstrap.），以及已编译压缩的 *CSS* 和 *JS*（*bootstrap.min.*）。同时也包含了 Glyphicons 的字体，这是一个可选的 Bootstrap 主题。

Bootstrap 源代码

如果您下载了 Bootstrap 源代码，那么文件结构将如下所示：



- *less/*、*js/* 和 *fonts/* 下的文件分别是 Bootstrap CSS、JS 和图标字体的源代码。
- *dist/* 文件夹包含了上面预编译下载部分中所列的文件和文件夹。
- *docs-assets/*、*examples/* 和所有的 **.html* 文件是 Bootstrap 文档。

HTML 模板

一个使用了 Bootstrap 的基本的 HTML 模板如下所示：

```
<!DOCTYPE html>
<html>
   <head>
      <title>Bootstrap 模板</title>
      <meta name="viewport" content="width=device-width, initial-sc
      <!-- 引入 Bootstrap -->
      <link href="css/bootstrap.min.css" rel="stylesheet">

      <!-- HTML5 Shim 和 Respond.js 用于让 IE8 支持 HTML5元素和媒体查询
      <!-- 注意： 如果通过 file://  引入 Respond.js 文件，则该文件无法起效
      <!--[if lt IE 9]>
         <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/h
         <script src="https://oss.maxcdn.com/libs/respond.js/1.3.0/
      <![endif]-->
   </head>
   <body>
      <h1>Hello, world!</h1>

      <!-- jQuery (Bootstrap 的 JavaScript 插件需要引入 jQuery) -->
      <script src="https://code.jquery.com/jquery.js"></script>
      <!-- 包括所有已编译的插件 -->
      <script src="js/bootstrap.min.js"></script>
   </body>
</html>
```

在这里，您可以看到包含了 **jquery.js**、**bootstrap.min.js** 和 **bootstrap.min.css** 文件，用于让一个常规的 HTML 文件变为使用了 Bootstrap 的模板。

有关上面代码段中每个元素的细节将在 Bootstrap CSS 概览 章节详细讲解。

实例

现在让我们尝试使用使用Bootstrap输出"Hello, world!"：

```
<!DOCTYPE html>
<html>
<head>
   <title>在线尝试 Bootstrap 实例</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

      <h1>Hello, world!</h1>

</body>
</html>
```

Bootstrap CDN推荐

本站实例采用的是百度的静态资源库(http://cdn.code.baidu.com/)上的Bootstrap资源。

百度的静态资源库的 CDN 服务，访问速度更快、加速效果更明显、没有速度和带宽限制、永久免费,引入代码如下：

```
<!-- 新 Bootstrap 核心 CSS 文件 -->
<link href="http://apps.bdimg.com/libs/bootstrap/3.0.3/css/bootstra

<!-- 可选的Bootstrap主题文件（一般不使用） -->
<script src="http://apps.bdimg.com/libs/bootstrap/3.0.3/css/bootst

<!-- jQuery文件。务必在bootstrap.min.js 之前引入 -->
<script src="http://apps.bdimg.com/libs/jquery/2.0.0/jquery.min.js'

<!-- 最新的 Bootstrap 核心 JavaScript 文件 -->
<script src="http://apps.bdimg.com/libs/bootstrap/3.0.3/js/bootstra
```

Bootstrap CSS

Bootstrap CSS 概览

在这一章中，我们将讲解 Bootstrap 底层结构的关键部分，包括我们让 web 开发变得更好、更快、更强壮的最佳实践。

HTML 5 文档类型（Doctype）

Bootstrap 使用了一些 HTML5 元素和 CSS 属性。为了让这些正常工作，您需要使用 HTML5 文档类型（Doctype）。 因此，请在使用 Bootstrap 项目的开头包含下面的代码段。

```
<!DOCTYPE html>
<html>
....
</html>
```

如果在 Bootstrap 创建的网页开头不使用 HTML5 的文档类型（Doctype），您可能会面临一些浏览器显示不一致的问题，甚至可能面临一些特定情境下的不一致，以致于您的代码不能通过 W3C 标准的验证。

移动设备优先

移动设备优先是 Bootstrap 3 的最显著的变化。

在之前的 Bootstrap 版本中（直到 2.x），您需要手动引用另一个 CSS，才能让整个项目友好的支持移动设备。

现在不一样了，Bootstrap 3 默认的 CSS 本身就对移动设备友好支持。

Bootstrap 3 的设计目标是移动设备优先，然后才是桌面设备。这实际上是一个非常及时的转变，因为现在越来越多的用户使用移动设备。

为了让 Bootstrap 开发的网站对移动设备友好，确保适当的绘制和触屏缩放，需要在网页的 head 之中添加 **viewport meta** 标签，如下所示：

```
<meta name="viewport" content="width=device-width, initial-scale=1
```

width 属性控制设备的宽度。假设您的网站将被带有不同屏幕分辨率的设备浏览，那么将它设置为 *device-width* 可以确保它能正确呈现在不同设备上。

initial-scale=1.0 确保网页加载时，以 1:1 的比例呈现，不会有任何的缩放。

在移动设备浏览器上，通过为 **viewport meta** 标签添加 *user-scalable=no* 可以禁用其缩放（zooming）功能。

通常情况下，*maximum-scale=1.0* 与 user-scalable=no 一起使用。这样禁用缩放功能后，用户只能滚动屏幕，就能让您的网站看上去更像原生应用的感觉。

注意，这种方式我们并不推荐所有网站使用，还是要看您自己的情况而定！

```
<meta name="viewport" content="width=device-width,
                                     initial-scale=1.0,
                                     maximum-scale=1.0,
                                     user-scalable=no">
```

响应式图像

```
<img src="..." class="img-responsive" alt="响应式图像">
```

通过添加 *img-responsive* class 可以让 Bootstrap 3 中的图像对响应式布局的支持更友好。

接下来让我们看下这个 class 包含了哪些 css 属性。

在下面的代码中，可以看到*img-responsive* class 为图像赋予了 max-width: 100%; 和 height: auto; 属性，可以让图像按比例缩放，不超过其父元素的尺寸。

```
.img-responsive {
  display: inline-block;
  height: auto;
  max-width: 100%;
}
```

这表明相关的图像呈现为 *inline-block*。当您把元素的 display 属性设置为 inline-block，元素相对于它周围的内容以内联形式呈现，但与内联不同的是，这种情况下我们可以设置宽度和高度。

设置 *height:auto*，相关元素的高度取决于浏览器。

设置 *max-width* 为 100% 会重写任何通过 width 属性指定的宽度。这让图片对响应式布局的支持更友好。

全局显示、排版和链接

基本的全局显示

Bootstrap 3 使用 *body {margin: 0;}* 来移除 body 的边距。

请看下面有关 body 的设置：

```
body {
  font-family: "Helvetica Neue", Helvetica, Arial, sans-serif;
  font-size: 14px;
  line-height: 1.428571429;
  color: #333333;
  background-color: #ffffff;
}
```

第一条规则设置 body 的默认字体样式为 *"Helvetica Neue", Helvetica, Arial, sans-serif*。

第二条规则设置文本的默认字体大小为 14 像素。

第三条规则设置默认的行高度为 1.428571429。

第四条规则设置默认的文本颜色为 #333333。

最后一条规则设置默认的背景颜色为白色。

排版

使用 @font-family-base、 @font-size-base 和 @line-height-base 属性作为排版样式。

链接样式

通过属性 @link-color 设置全局链接的颜色。

对于链接的默认样式，如下设置：

```
a:hover,
a:focus {
  color: #2a6496;
  text-decoration: underline;
}

a:focus {
  outline: thin dotted #333;
  outline: 5px auto -webkit-focus-ring-color;
  outline-offset: -2px;
}
```

所以，当鼠标悬停在链接上，或者点击过的链接，颜色会被设置为 #2a6496。同时，会呈现一条下划线。

除此之外，点击过的链接，会呈现一个颜色码为 #333 的细的虚线轮廓。另一条规则是设置轮廓为 5 像素宽，且对于基于 webkit 浏览器有一个 *-webkit-focus-ring-color* 的浏览器扩展。轮廓偏移设置为 -2 像素。

以上所有这些样式都可以在 scaffolding.less 中找到。

避免跨浏览器的不一致

Bootstrap 使用 Normalize 来建立跨浏览器的一致性。

Normalize.css 是一个很小的 CSS 文件，在 HTML 元素的默认样式中提供了更好的跨浏览器一致性。

容器（Container）

```
<div class="container">
  ...
</div>
```

Bootstrap 3 的 *container* class 用于包裹页面上的内容。让我们一起来看看 bootstrap.css 文件中的这个 *.container* class。

```
.container {
   padding-right: 15px;
   padding-left: 15px;
   margin-right: auto;
   margin-left: auto;
}
```

通过上面的代码，把 container 的左右外边距（margin-right、margin-left）交由浏览器决定。

请注意，由于内边距（padding）是固定宽度，默认情况下容器是不可嵌套的。

```
.container:before,
.container:after {
  display: table;
  content: " ";
}
```

这会产生伪元素。设置 *display* 为 *table*，会创建一个匿名的 table-cell 和一个新的块格式化上下文。*:before* 伪元素防止上边距崩塌，*:after* 伪元素清除浮动。

如果 *conteneditable* 属性出现在 HTML 中，由于一些 Opera bug，围绕上述元素创建一个空格。这可以通过使用 *content: " "* 来修复。

```
.container:after {
  clear: both;
}
```

它创建了一个伪元素，并确保了所有的容器包含所有的浮动元素。

Bootstrap 3 CSS 有一个申请响应的媒体查询，在不同的媒体查询阈值范围内都为 container 设置了max-width，用以匹配网格系统。

```
@media (min-width: 768px) {
   .container {
      width: 750px;
}
```

Bootstrap 浏览器/设备支持

Bootstrap 可以在最新的桌面系统和移动端浏览器中很好的工作。

旧的浏览器可能无法很好的支持。

下表为 Bootstrap 支持最新版本的浏览器和平台：

	Chrome	Firefox	IE	Opera	Safari
Android	YES	YES	不适用	NO	不适用
iOS	YES	不适用	不适用	NO	YES
Mac OS X	YES	YES	不适用	YES	YES
Windows	YES	YES	YES*	YES	NO

- Bootstrap 支持 Internet Explorer 8 及更高版本的 IE 浏览器。

Bootstrap 网格系统

本章节我们将讲解 Bootstrap 的网格系统（Grid System）。

Bootstrap 提供了一套响应式、移动设备优先的流式网格系统，随着屏幕或视口（viewport）尺寸的增加，系统会自动分为最多12列。

什么是网格（Grid）？

摘自维基百科：

> 在平面设计中，网格是一种由一系列用于组织内容的相交的直线（垂直的、水平的）组成的结构（通常是二维的）。它广泛应用于打印设计中的设计布局和内容结构。在网页设计中，它是一种用于快速创建一致的布局和有效地使用 HTML 和 CSS 的方法。

简单地说，网页设计中的网格用于组织内容，让网站易于浏览，并降低用户端的负载。

什么是 Bootstrap 网格系统（Grid System）？

Bootstrap 官方文档中有关网格系统的描述：

> Bootstrap 包含了一个响应式的、移动设备优先的、不固定的网格系统，可以随着设备或视口大小的增加而适当地扩展到 12 列。它包含了用于简单的布局选项的预定义类，也包含了用于生成更多语义布局的功能强大的混合类。

让我们来理解一下上面的语句。Bootstrap 3 是移动设备优先的，在这个意义上，Bootstrap 代码从小屏幕设备（比如移动设备、平板电脑）开始，然后扩展到大屏幕设备（比如笔记本电脑、台式电脑）上的组件和网格。

移动设备优先策略

- 内容
  - 决定什么是最重要的。
- 布局
  - 优先设计更小的宽度。
  - 基础的 CSS 是移动设备优先，媒体查询是针对于平板电脑、台式电脑。
- 渐进增强
  - 随着屏幕大小的增加而添加元素。

相应式的网格系统随着屏幕或视口（viewport）尺寸的增加，系统会自动分为最多12列。

1 1 1 1 1 1 1 1 1 1 1 1
4 4 4
4 8
6 6
12

Bootstrap 网格系统（Grid System）的工作原理

网格系统通过一系列包含内容的行和列来创建页面布局。下面列出了 Bootstrap 网格系统是如何工作的：

- 行必须放置在 **.container** class 内，以便获得适当的对齐（alignment）和内边距（padding）。
- 使用行来创建列的水平组。
- 内容应该放置在列内，且唯有列可以是行的直接子元素。
- 预定义的网格类，比如 **.row** 和 **.col-xs-4**，可用于快速创建网格布局。LESS 混合类可用于更多语义布局。
- 列通过内边距（padding）来创建列内容之间的间隙。该内边距是通过 **.rows** 上的外边距（margin）取负，表示第一列和最后一列的行偏移。
- 网格系统是通过指定您想要横跨的十二个可用的列来创建的。例如，要创建三个相等的列，则使用三个 **.col-xs-4**。

媒体查询

媒体查询是非常别致的"有条件的 CSS 规则"。它只适用于一些基于某些规定条件的 CSS。如果满足那些条件，则应用相应的样式。

Bootstrap 中的媒体查询允许您基于视口大小移动、显示并隐藏内容。下面的媒体查询在 LESS 文件中使用，用来创建 Bootstrap 网格系统中的关键的分界点阈值。

```
/* 超小设备（手机，小于 768px） */
/* Bootstrap 中默认情况下没有媒体查询 */

/* 小型设备（平板电脑，768px 起） */
@media (min-width: @screen-sm-min) { ... }

/* 中型设备（台式电脑，992px 起） */
@media (min-width: @screen-md-min) { ... }

/* 大型设备（大台式电脑，1200px 起） */
@media (min-width: @screen-lg-min) { ... }
```

我们有时候也会在媒体查询代码中包含 **max-width**，从而将 CSS 的影响限制在更小范围的屏幕大小之内。

```
@media (max-width: @screen-xs-max) { ... }
@media (min-width: @screen-sm-min) and (max-width: @screen-sm-max)
@media (min-width: @screen-md-min) and (max-width: @screen-md-max)
@media (min-width: @screen-lg-min) { ... }
```

媒体查询有两个部分，先是一个设备规范，然后是一个大小规则。在上面的案例中，设置了下列的规则：

让我们来看下面这行代码：

```
@media (min-width: @screen-sm-min) and (max-width: @screen-sm-max)
```

对于所有带有 *min-width: @screen-sm-min* 的设备，如果屏幕的宽度小于 *@screen-sm-max*，则会进行一些处理。

网格选项

下表总结了 Bootstrap 网格系统如何跨多个设备工作：

	超小设备收集（<768px）	小型设备平板电脑（≥768px）	中型设备台式电脑（≥992px）	大型设备台式电脑（≥1200px）
网格行为	一直是水平的	以折叠开始，断点以上是水平的	以折叠开始，断点以上是水平的	以折叠开始，断点以上是水平的
最大容器宽度	None (auto)	750px	970px	1170px
Class 前缀	**.col-xs-**	**.col-sm-**	**.col-md-**	**.col-lg-**
列 #	12	12	12	12
最大列宽	Auto	60px	78px	95px
间隙宽度	30px（一个列的每边分别 15px）	30px（一个列的每边分别 15px）	30px（一个列的每边分别 15px）	30px（一个列的每边分别 15px）
可嵌套	Yes	Yes	Yes	Yes
偏移量	Yes	Yes	Yes	Yes
列排序	Yes	Yes	Yes	Yes

基本的网格结构

下面是 Bootstrap 网格的基本结构：

```
<div class="container">
   <div class="row">
      <div class="col-*-*"></div>
      <div class="col-*-*"></div>
   </div>
   <div class="row">...</div>
</div>
<div class="container">....
```

让我们来看几个简单的网格实例：

- 实例：堆叠的水平
- 实例：中型和大型设备

- 实例：手机、平板电脑、台式电脑

响应式的列重置

以下实例包含了4个网格，但是我们在小设备浏览时无法确定网格显示的位置。

为了解决这个问题，使用 可以使用 **.clearfix** class和 响应式实用工具来解决，如下面实例所示：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 响应式的列重置</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="container">
   <div class="row" >
      <div class="col-xs-6 col-sm-3"
         style="background-color: #dedef8;
         box-shadow: inset 1px -1px 1px #444, inset -1px 1px 1px #4
         <p>Lorem ipsum dolor sit amet, consectetur adipisicing eli
      </div>
      <div class="col-xs-6 col-sm-3"
         style="background-color: #dedef8;box-shadow:
         inset 1px -1px 1px #444, inset -1px 1px 1px #444;">
         <p>Lorem ipsum dolor sit amet, consectetur adipisicing eli
            eiusmod tempor incididunt ut labore et dolore magna ali
            enim ad minim veniam, quis nostrud exercitation ullamco
            nisi ut aliquip ex ea commodo consequat.
         </p>
         <p>Lorem ipsum dolor sit amet, consectetur adipisicing eli
            eiusmod tempor incididunt ut.
         </p>
      </div>

      <div class="clearfix visible-xs"></div>

      <div class="col-xs-6 col-sm-3"
         style="background-color: #dedef8;
         box-shadow:inset 1px -1px 1px #444, inset -1px 1px 1px #44
         <p>Ut enim ad minim veniam, quis nostrud exercitation ulla
            laboris nisi ut aliquip ex ea commodo consequat.
         </p>
      </div>
      <div class="col-xs-6 col-sm-3"
         style="background-color: #dedef8;box-shadow:
         inset 1px -1px 1px #444, inset -1px 1px 1px #444;">
```

```
          <p>Lorem ipsum dolor sit amet, consectetur adipisicing el
             eiusmod tempor incididunt ut labore et dolore magna al
             enim ad minim
          </p>
       </div>
    </div>
</div>

</body>
</html>
```

浏览器上调整窗口大小查看变化，或在您手机上查看效果。

偏移列

偏移是一个用于更专业的布局的有用功能。它们可用来给列腾出更多的空间。例如，**.col-xs=*** 类不支持偏移，但是它们可以简单地通过使用一个空的单元格来实现该效果。

为了在大屏幕显示器上使用偏移，请使用 **.col-md-offset-*** 类。这些类会把一个列的左外边距（margin）增加 * 列，其中 * 范围是从 **1** 到 **11**。

在下面的实例中，我们有 <div class="col-md-6">..</div>，我们将使用 **.col-md-offset-3** class 来居中这个 div。

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 偏移列</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="container">

   <h1>Hello, world!</h1>

   <div class="row" >
      <div class="col-xs-6 col-md-offset-3"
         style="background-color: #dedef8;box-shadow:
         inset 1px -1px 1px #444, inset -1px 1px 1px #444;">
         <p>Lorem ipsum dolor sit amet, consectetur adipisicing
            elit.
         </p>
      </div>

   </div>
</div>

</body>
</html>
```

结果如下所示：

Hello, world!

Lorem ipsum dolor sit amet, consectetur adipisicing elit.

嵌套列

为了在内容中嵌套默认的网格，请添加一个新的 **.row**，并在一个已有的 **.col-md-*** 列内添加一组 **.col-md-*** 列。被嵌套的行应包含一组列，这组列个数不能超过 12（其实，没有要求你必须占满12列）。

在下面的实例中，布局有两个列，第二列被分为两行四个盒子。

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 嵌套列</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
```

```
    <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="container">

    <h1>Hello, world!</h1>

    <div class="row">

        <div class="col-md-3" style="background-color: #dedef8;box-sh
            inset 1px -1px 1px #444, inset -1px 1px 1px #444;">
            <h4>第一列</h4>
            <p>Lorem ipsum dolor sit amet, consectetur adipisicing eli
        </div>

        <div class="col-md-9" style="background-color: #dedef8;box-sh
            inset 1px -1px 1px #444, inset -1px 1px 1px #444;">
            <h4>第二列 - 分为四个盒子</h4>
            <div class="row">
                <div class="col-md-6" style="background-color: #B18904;
                    box-shadow: inset 1px -1px 1px #444, inset -1px 1px
                    <p>Consectetur art party Tonx culpa semiotics. Pinte
                        assumenda minim organic quis.
                    </p>
                </div>
                <div class="col-md-6" style="background-color: #B18904;
                    box-shadow: inset 1px -1px 1px #444, inset -1px 1px
                    <p> sed do eiusmod tempor incididunt ut labore et do
                        aliqua. Ut enim ad minim veniam, quis nostrud exe
                        ullamco laboris nisi ut aliquip ex ea commodo con
                    </p>
                </div>
            </div>

            <div class="row">
                <div class="col-md-6" style="background-color: #B18904;
                    box-shadow: inset 1px -1px 1px #444, inset -1px 1px
                    <p>quis nostrud exercitation ullamco laboris nisi ut
                        aliquip ex ea commodo consequat.
                    </p>
                </div>
                <div class="col-md-6" style="background-color: #B18904;
                    box-shadow: inset 1px -1px 1px #444, inset -1px 1px
                    <p>Lorem ipsum dolor sit amet, consectetur adipisici
                        sed do eiusmod tempor incididunt ut labore et dol
                        aliqua. Ut enim ad minim.</p>
                </div>
            </div>

        </div>

    </div>
```

```
</div>

</body>
</html>
```

结果如下所示：

Hello, world!

第一列

Lorem ipsum dolor sit amet, consectetur adipisicing elit.

第二列 - 分为四个盒子

Consectetur art party Tonx culpa semiotics. Pinterest assumenda minim organic quis.

sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim.

列排序

Bootstrap 网格系统另一个完美的特性，就是您可以很容易地以一种顺序编写列，然后以另一种顺序显示列。

您可以很轻易地改变带有 **.col-md-push-*** 和 **.col-md-pull-*** 类的内置网格列的顺序，其中 * 范围是从 **1** 到 **11**。

在下面的实例中，我们有两列布局，左列很窄，作为侧边栏。我们将使用 **.col-md-push-*** 和 **.col-md-pull-*** 类来互换这两列的顺序。

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 列排序</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="container">

   <h1>Hello, world!</h1>

   <div class="row">
      <p>排序前</p>
      <div class="col-md-4" style="background-color: #dedef8;
         box-shadow: inset 1px -1px 1px #444, inset -1px 1px 1px #4
         我在左边
      </div>
      <div class="col-md-8" style="background-color: #dedef8;
         box-shadow: inset 1px -1px 1px #444, inset -1px 1px 1px #4
         我在右边
      </div>
   </div><br>
   <div class="row">
      <p>排序后</p>
      <div class="col-md-4 col-md-push-8"
         style="background-color: #dedef8;
         box-shadow: inset 1px -1px 1px #444,
         inset -1px 1px 1px #444;">
         我在左边
      </div>
      <div class="col-md-8 col-md-pull-4"
         style="background-color: #dedef8;
         box-shadow: inset 1px -1px 1px #444,
         inset -1px 1px 1px #444;">
         我在右边
      </div>
   </div>

</div>

</body>
</html>
```

结果如下所示：

Hello, world!

排序前

我在左边	我在右边

排序后

我在右边	我在左边

Bootstrap 排版

Bootstrap 使用 Helvetica Neue、 Helvetica、 Arial 和 sans-serif 作为其默认的字体栈。

使用 Bootstrap 的排版特性，您可以创建标题、段落、列表及其他内联元素。

标题

Bootstrap 中定义了所有的 HTML 标题（h1 到 h6）的样式。请看下面的实例：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 标题</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<h1>我是标题1 h1</h1>
<h2>我是标题2 h2</h2>
<h3>我是标题3 h3</h3>
<h4>我是标题4 h4</h4>
<h5>我是标题5 h5</h5>
<h6>我是标题6 h6</h6>

</body>
</html>
```

结果如下所示：

我是标题1 h1

我是标题2 h2

我是标题3 h3

我是标题4 h4

我是标题5 h5

我是标题6 h6

内联子标题

如果需要向任何标题添加一个内联子标题，只需要简单地在元素两旁添加 <small>，或者添加 **.small** class，这样子您就能得到一个字号更小的颜色更浅的文本，如下面实例所示：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 内联子标题</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<h1>我是标题1 h1\. <small>我是副标题1 h1</small></h1>
<h2>我是标题2 h2\. <small>我是副标题2 h2</small></h2>
<h3>我是标题3 h3\. <small>我是副标题3 h3</small></h3>
<h4>我是标题4 h4\. <small>我是副标题4 h4</small></h4>
<h5>我是标题5 h5\. <small>我是副标题5 h5</small></h5>
<h6>我是标题6 h6\. <small>我是副标题6 h6</small></h6>

</body>
</html>
```

结果如下所示：

我是标题1 h1. 我是副标题1 h1

我是标题2 h2. 我是副标题2 h2

我是标题3 h3. 我是副标题3 h3

我是标题4 h4. 我是副标题4 h4

我是标题5 h5. 我是副标题5 h5

我是标题6 h6. 我是副标题6 h6

引导主体副本

为了给段落添加强调文本，则可以添加 class="lead"，这将得到更大更粗、行高更高的文本，如下面实例所示：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 引导主体副本</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>
<hr>
<h2>引导主体副本</h2>
<p class="lead">这是一个演示引导主体副本用法的实例。这是一个演示引导主体副本

</body>
</html>
```

结果如下所示：

引导主体副本

这是一个演示引导主体副本用法的实例。这是一个演示引导主体副本用法的实例。这是一个演示引导主体副本用法的实例。这是一个演示引导主体副本用法的实例。这是一个演示引导主体副本用法的实例。这是一个演示引导主体副本用法的实例。这是一个演示引导主体副本用法的实例。这是一个演示引导主体副本用法的实例。

强调

HTML 的默认强调标签 <small>（设置文本为父文本大小的 85%）、<strong>（设置文本为更粗的文本）、<em>（设置文本为斜体）。

Bootstrap 提供了一些用于强调文本的类，如下面实例所示：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 强调</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<small>本行内容是在标签内</small><br>
<strong>本行内容是在标签内</strong><br>
<em>本行内容是在标签内，并呈现为斜体</em><br>
<p class="text-left">向左对齐文本</p>
<p class="text-center">居中对齐文本</p>
<p class="text-right">向右对齐文本</p>
<p class="text-muted">本行内容是减弱的</p>
<p class="text-primary">本行内容带有一个 warning class</p>
<p class="text-success">本行内容带有一个 success class</p>
<p class="text-info">本行内容带有一个 info class</p>
<p class="text-warning">本行内容带有一个 warning class</p>
<p class="text-danger">本行内容带有一个 danger class</p>

</body>
</html>
```

结果如下所示：



缩写

HTML <abbr>元素提供了用于缩写的标记，比如 WWW 或 HTTP。Bootstrap 定义 <abbr> 元素的样式为显示在文本底部的一条虚线边框，当鼠标悬停在上面时会显示完整的文本（只要您为 <abbr> title 属性添加了文本）。为了得到一个更小字体的文本，请添加 .initialism 到 <abbr>。</abbr>

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 缩写</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<abbr title="World Wide Web">WWW</abbr><br>
<abbr title="Real Simple Syndication" class="initialism">RSS</abbr>

</body>
</html>
```

结果如下所示：



地址（Address）

使用 <address> 标签，您可以在网页上显示联系信息。由于 <address> 默认为 display:block;，您需要使用 标签来为封闭的地址文本添加换行。

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 地址</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<address>
  <strong>Some Company, Inc.</strong><br>
  007 street<br>
  Some City, State XXXXX<br>
  <abbr title="Phone">P:</abbr> (123) 456-7890
</address>

<address>
  <strong>Full Name</strong><br>
  <a href="mailto:#">mailto@somedomain.com</a>
</address>

</body>
</html>
```

结果如下所示：

Some Company, Inc.
007 street
Some City, State XXXXX
P: (123) 456-7890

Full Name
mailto@somedomain.com

引用（Blockquote）

您可以在任意的 HTML 文本旁使用默认的 <blockquote>。其他选项包括，添加一个 <small> 标签来标识引用的来源，使用 class *.pull-right* 向右对齐引用。下面的实例演示了这些特性：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 引用</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<blockquote><p>
这是一个默认的引用实例。这是一个默认的引用实例。这是一个默认的引用实例。这是一个
<blockquote>这是一个带有源标题的引用。<small>Someone famous in <cite ti
<blockquote class="pull-right">这是一个向右对齐的引用。<small>Someone f

</body>
</html>
```

结果如下所示：

这是一个默认的引用实例。这是一个默认的引用实例。这是一个默认的引用实例。
这是一个默认的引用实例。这是一个默认的引用实例。这是一个默认的引用实例。
这是一个默认的引用实例。这是一个默认的引用实例。

这是一个带有源标题的引用。
— Someone famous in Source Title

这是一个向右对齐的引用。
Someone famous in Source Title —

列表

Bootstrap 支持有序列表、无序列表和定义列表。

- 有序列表：有序列表是指以数字或其他有序字符开头的列表。
- 无序列表：无序列表是指没有特定顺序的列表，是以传统风格的着重号开头的列表。如果您不想显示这些着重号，您可以使用 class *.list-unstyled* 来移除样式。您也可以通过使用 class *.list-inline* 把所有的列表项放在同一行中。
- 定义列表：在这种类型的列表中，每个列表项可以包含 <dt> 和 <dd> 元素。<dt> 代表 定义术语，就像字典，这是被定义的属于（或短语）。接着，<dd> 是 <dt> 的描述。您可以使用 class *dl-horizontal* 把 <dl> 行中的属于与描述并排显示。

下面的实例演示了这些类型的列表：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 列表</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<h4>有序列表</h4>
<ol>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
  <li>Item 4</li>
</ol>
<h4>无序列表</h4>
<ul>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
  <li>Item 4</li>
</ul>
<h4>未定义样式列表</h4>
<ul class="list-unstyled">
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
  <li>Item 4</li>
</ul>
<h4>内联列表</h4>
<ul class="list-inline">
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
  <li>Item 4</li>
</ul>
<h4>定义列表</h4>
<dl>
  <dt>Description 1</dt>
  <dd>Item 1</dd>
  <dt>Description 2</dt>
  <dd>Item 2</dd>
</dl>
<h4>水平的定义列表</h4>
<dl class="dl-horizontal">
  <dt>Description 1</dt>
  <dd>Item 1</dd>
  <dt>Description 2</dt>
  <dd>Item 2</dd>
</dl>
```

```
</body>
</html>
```

结果如下所示：

有序列表

1. Item 1
2. Item 2
3. Item 3
4. Item 4

无序列表

- Item 1
- Item 2
- Item 3
- Item 4

未定义样式列表

Item 1
Item 2
Item 3
Item 4

内联列表

Item 1 Item 2 Item 3 Item 4

定义列表

Description 1
Item 1
Description 2
Item 2

水平的定义列表

Description 1 Item 1
Description 2 Item 2

更多排版类

下表提供了 Bootstrap 更多排版类的实例：

类	描述	实例
.lead	使段落突出显示	尝试一下
.small	设定小文本 (设置为父文本的 85% 大小)	尝试

.small	设定小文本 (设置为父文本的 85% 大小)	一下
.text-left	设定文本左对齐	尝试一下
.text-center	设定文本居中对齐	尝试一下
.text-right	设定文本右对齐	尝试一下
.text-justify	设定文本对齐,段落中超出屏幕部分文字自动换行	尝试一下
.text-nowrap	段落中超出屏幕部分不换行	尝试一下
.text-lowercase	设定文本小写	尝试一下
.text-uppercase	设定文本大写	尝试一下
.text-capitalize	设定单词首字母大写	尝试一下
.initialism	显示在 <abbr> 元素中的文本以小号字体展示	尝试一下
.blockquote-reverse	设定引用右对齐	尝试一

.list-unstyled	移除默认的列表样式，列表项中左对齐 (和 中)。 这个类仅适用于直接子列表项 (如果需要移除嵌套的列表项，你需要在嵌套的列表中使用该样式)	尝试一下
.list-inline	将所有列表项放置同一行	尝试一下
.dl-horizontal	该类设置了浮动和偏移，应用于 <dl> 元素和 <dt> 元素中，具体实现可以查看实例	尝试一下
.pre-scrollable	使 <pre> 元素可滚动 scrollable	尝试一下

Bootstrap 代码

Bootstrap 允许您以两种方式显示代码：

- 第一种是 <code> 标签。如果您想要内联显示代码，那么您应该使用 <code> 标签。
- 第二种是 <pre> 标签。如果代码需要被显示为一个独立的块元素或者代码有多行，那么您应该使用 <pre> 标签。

请确保当您使用 <pre> 和 <code> 标签时，开始和结束标签使用了 unicode 变体：**<** 和 **>**。

让我们来看看下面的实例：

实例

```
<!DOCTYPE html>
<html>
<head>
  <title>Bootstrap 实例 - 代码</title>
  <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
  <script src="/scripts/jquery.min.js"></script>
  <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<p><code>&lt;header&gt;</code> 作为内联元素被包围。</p>
<p>如果需要把代码显示为一个独立的块元素，请使用 <pre> 标签：</p>
<pre>
  &lt;article&gt;
    &lt;h1&gt;Article Heading&lt;/h1&gt;
  &lt;/article&gt;
</pre>
</body>
</html>
```

实例展示如下图：

<header> 作为内联元素被包围。

如果需要把代码显示为一个独立的块元素，请使用 <pre> 标签：

```
<article>
  <h1>Article Heading</h1>
</article>
```

Bootstrap 表格

Bootstrap 提供了一个清晰的创建表格的布局。下表列出了 Bootstrap 支持的一些表格元素：

标签	描述
<table>	容纳以表格形式显示数据的元素。
<thead>	表格标题行的容器元素（<tr>），用来标识表格列。
<tbody>	表格主体中的表格行的容器元素（<tr>）。
<tr>	一组出现在单行上的表格单元格的容器元素（<td> 或 <th>）。
<td>	默认的表格单元格。
<th>	特殊的表格单元格，用来标识列或行（取决于范围和位置）。必须在 <thead> 内使用。
<caption>	关于表格存储内容的描述或总结。

基本的表格

如果您想要一个只带有内边距（padding）和水平分割的基本表，请添加 class *.table*，如下面实例所示：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 基本的表格</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<table class="table">
   <caption>基本的表格布局</caption>
   <thead>
      <tr>
         <th>名称</th>
         <th>城市</th>
      </tr>
   </thead>
   <tbody>
      <tr>
         <td>Tanmay</td>
         <td>Bangalore</td>
      </tr>
      <tr>
         <td>Sachin</td>
         <td>Mumbai</td>
      </tr>
   </tbody>
</table>

</body>
</html>
```

结果如下所示：

基本的表格布局

名称	城市
Tanmay	Bangalore
Sachin	Mumbai

可选的表格类

除了基本的表格标记和 .table class，还有一些可以用来为标记定义样式的类。下面将向您介绍这些类。

条纹表格

通过添加 *.table-striped* class，您将在 <tbody> 内的行上看到条纹，如下面的实例所示：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 条纹表格</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<table class="table table-striped">
   <caption>条纹表格布局</caption>
   <thead>
      <tr>
         <th>名称</th>
         <th>城市</th>
         <th>密码</th>
      </tr>
   </thead>
   <tbody>
      <tr>
         <td>Tanmay</td>
         <td>Bangalore</td>
         <td>560001</td>
      </tr>
      <tr>
         <td>Sachin</td>
         <td>Mumbai</td>
         <td>400003</td>
      </tr>
      <tr>
         <td>Uma</td>
         <td>Pune</td>
         <td>411027</td>
      </tr>
   </tbody>
</table>

</body>
</html>
```

结果如下所示：

名称	城市	密码
Tanmay	Bangalore	560001
Sachin	Mumbai	400003
Uma	Pune	411027

条纹表格布局

边框表格

通过添加 *.table-bordered* class，您将看到每个元素周围都有边框，且占整个表格是圆角的，如下面的实例所示：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 边框表格</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<table class="table table-bordered">
   <caption>边框表格布局</caption>
   <thead>
      <tr>
         <th>名称</th>
         <th>城市</th>
         <th>密码</th>
      </tr>
   </thead>
   <tbody>
      <tr>
         <td>Tanmay</td>
         <td>Bangalore</td>
         <td>560001</td>
      </tr>
      <tr>
         <td>Sachin</td>
         <td>Mumbai</td>
         <td>400003</td>
      </tr>
      <tr>
         <td>Uma</td>
         <td>Pune</td>
         <td>411027</td>
      </tr>
   </tbody>
</table>

</body>
</html>
```

结果如下所示：

边框表格布局

名称	城市	密码
Tanmay	Bangalore	560001
Sachin	Mumbai	400003
Uma	Pune	411027

悬停表格

通过添加 *.table-hover* class，当指针悬停在行上时会出现浅灰色背景，如下面的实例所示：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 悬停表格</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<table class="table table-hover">
   <caption>悬停表格布局</caption>
   <thead>
      <tr>
         <th>名称</th>
         <th>城市</th>
         <th>密码</th>
      </tr>
   </thead>
   <tbody>
      <tr>
         <td>Tanmay</td>
         <td>Bangalore</td>
         <td>560001</td>
      </tr>
      <tr>
         <td>Sachin</td>
         <td>Mumbai</td>
         <td>400003</td>
      </tr>
      <tr>
         <td>Uma</td>
         <td>Pune</td>
         <td>411027</td>
      </tr>
   </tbody>
</table>

</body>
</html>
```

结果如下所示：

悬停表格布局

名称	城市	密码
Tanmay	Bangalore	560001
Sachin	Mumbai	400003
Uma	Pune	411027

精简表格

通过添加 *.table-condensed* class，行内边距（padding）被切为两半，以便让表看起来更紧凑，如下面的实例所示。这在想让信息看起来更紧凑时非常有用。

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 精简表格</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<table class="table table-condensed">
   <caption>精简表格布局</caption>
   <thead>
      <tr>
         <th>名称</th>
         <th>城市</th>
         <th>密码</th>
      </tr>
   </thead>
   <tbody>
      <tr>
         <td>Tanmay</td>
         <td>Bangalore</td>
         <td>560001</td>
      </tr>
      <tr>
         <td>Sachin</td>
         <td>Mumbai</td>
         <td>400003</td>
      </tr>
      <tr>
         <td>Uma</td>
         <td>Pune</td>
         <td>411027</td>
      </tr>
   </tbody>
</table>

</body>
</html>
```

结果如下所示：

精简表格布局

名称	城市	密码
Tanmay	Bangalore	560001
Sachin	Mumbai	400003
Uma	Pune	411027

上下文类

下表中所列出的上下文类允许您改变表格行或单个单元格的背景颜色。

类	描述
.active	对某一特定的行或单元格应用悬停颜色
.success	表示一个成功的或积极的动作
.warning	表示一个需要注意的警告
.danger	表示一个危险的或潜在的负面动作

这些类可被应用到 <tr>、<td> 或 <th>。

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 上下文类</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<table class="table">
   <caption>上下文表格布局</caption>
   <thead>
      <tr>
         <th>产品</th>
         <th>付款日期</th>
         <th>状态</th>
      </tr>
   </thead>
   <tbody>
      <tr class="active">
         <td>产品1</td>
         <td>23/11/2013</td>
         <td>待发货</td>
      </tr>
      <tr class="success">
         <td>产品2</td>
         <td>10/11/2013</td>
         <td>发货中</td>
      </tr>
      <tr  class="warning">
         <td>产品3</td>
         <td>20/10/2013</td>
         <td>待确认</td>
      </tr>
      <tr  class="danger">
         <td>产品4</td>
         <td>20/10/2013</td>
         <td>已退货</td>
      </tr>
   </tbody>
</table>

</body>
</html>
```

结果如下所示：

上下文表格布局

产品	付款日期	状态
产品1	23/11/2013	待发货
产品2	10/11/2013	发货中
产品3	20/10/2013	待确认
产品4	20/10/2013	已退货

响应式表格

通过把任意的 *.table* 包在 *.table-responsive* class 内，您可以让表格水平滚动以适应小型设备（小于 768px）。当在大于 768px 宽的大型设备上查看时，您将看不到任何的差别。

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 响应式表格</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="table-responsive">
   <table class="table">
      <caption>响应式表格布局</caption>
      <thead>
         <tr>
            <th>产品</th>
            <th>付款日期</th>
            <th>状态</th>
         </tr>
      </thead>
      <tbody>
         <tr>
            <td>产品1</td>
            <td>23/11/2013</td>
            <td>待发货</td>
         </tr>
         <tr>
            <td>产品2</td>
            <td>10/11/2013</td>
            <td>发货中</td>
         </tr>
         <tr>
            <td>产品3</td>
            <td>20/10/2013</td>
            <td>待确认</td>
         </tr>
         <tr>
            <td>产品4</td>
            <td>20/10/2013</td>
            <td>已退货</td>
         </tr>
      </tbody>
   </table>
</div>

</body>
</html>
```

结果如下所示：

响应式表格布局

产品	付款日期	状态
产品1	23/11/2013	待发货
产品2	10/11/2013	发货中
产品3	20/10/2013	待确认
产品4	20/10/2013	已退货

Bootstrap 表单

在本章中，我们将学习如何使用 Bootstrap 创建表单。Bootstrap 通过一些简单的 HTML 标记和扩展的类即可创建出不同样式的表单。

表单布局

Bootstrap 提供了下列类型的表单布局：

- 垂直表单（默认）
- 内联表单
- 水平表单

垂直或基本表单

基本的表单结构是 Bootstrap 自带的，个别的表单控件自动接收一些全局样式。下面列出了创建基本表单的步骤：

- 向父 <form> 元素添加 *role="form"*。
- 把标签和控件放在一个带有 class *.form-group* 的 <div> 中。这是获取最佳间距所必需的。
- 向所有的文本元素 <input>、<textarea> 和 <select> 添加 class *.form-control*。

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 基本表单</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<form role="form">
   <div class="form-group">
      <label for="name">名称</label>
      <input type="text" class="form-control" id="name"
         placeholder="请输入名称">
   </div>
   <div class="form-group">
      <label for="inputfile">文件输入</label>
      <input type="file" id="inputfile">
      <p class="help-block">这里是块级帮助文本的实例。</p>
   </div>
   <div class="checkbox">
      <label>
      <input type="checkbox"> 请打勾
      </label>
   </div>
   <button type="submit" class="btn btn-default">提交</button>
</form>

</body>
</html>
```

结果如下所示：

名称

请输入名称

文件输入

选择文件 未选择文件

这里是块级帮助文本的实例。

☐ 请打勾

提交

内联表单

如果需要创建一个表单，它的所有元素是内联的，向左对齐的，标签是并排的，请向 <form> 标签添加 class *.form-inline*。

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 内联表单</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<form class="form-inline" role="form">
   <div class="form-group">
      <label class="sr-only" for="name">名称</label>
      <input type="text" class="form-control" id="name"
         placeholder="请输入名称">
   </div>
   <div class="form-group">
      <label class="sr-only" for="inputfile">文件输入</label>
      <input type="file" id="inputfile">
   </div>
   <div class="checkbox">
      <label>
      <input type="checkbox"> 请打勾
      </label>
   </div>
   <button type="submit" class="btn btn-default">提交</button>
</form>

</body>
</html>
```

结果如下所示：

请输入名称 浏览… 未选择文件。 ☐ 请打勾 提交

- 默认情况下，Bootstrap 中的 input、select 和 textarea 有 100% 宽度。在使用内联表单时，您需要在表单控件上设置一个宽度。
- 使用 class *.sr-only*，您可以隐藏内联表单的标签。

水平表单

水平表单与其他表单不仅标记的数量上不同，而且表单的呈现形式也不同。如需创建一个水平布局的表单，请按下面的几个步骤进行：

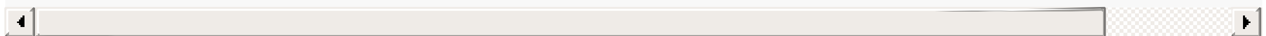
- 向父 <form> 元素添加 class *.form-horizontal*。

- 把标签和控件放在一个带有 class *.form-group* 的 <div> 中。
- 向标签添加 class *.control-label*。

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 水平表单</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<form class="form-horizontal" role="form">
   <div class="form-group">
      <label for="firstname" class="col-sm-2 control-label">名字</l
      <div class="col-sm-10">
         <input type="text" class="form-control" id="firstname"
            placeholder="请输入名字">
      </div>
   </div>
   <div class="form-group">
      <label for="lastname" class="col-sm-2 control-label">姓</labe
      <div class="col-sm-10">
         <input type="text" class="form-control" id="lastname"
            placeholder="请输入姓">
      </div>
   </div>
   <div class="form-group">
      <div class="col-sm-offset-2 col-sm-10">
         <div class="checkbox">
            <label>
               <input type="checkbox"> 请记住我
            </label>
         </div>
      </div>
   </div>
   <div class="form-group">
      <div class="col-sm-offset-2 col-sm-10">
         <button type="submit" class="btn btn-default">登录</button
      </div>
   </div>
</form>

</body>
</html>
```



结果如下所示：

名字 请输入名字

姓 请输入姓

☐ 请记住我

登录

支持的表单控件

Bootstrap 支持最常见的表单控件，主要是 *input*、*textarea*、*checkbox*、*radio* 和 *select*。

输入框（Input）

最常见的表单文本字段是输入框 input。用户可以在其中输入大多数必要的表单数据。Bootstrap 提供了对所有原生的 HTML5 的 input 类型的支持，包括：*text*、*password*、*datetime*、*datetime-local*、*date*、*month*、*time*、*week*、*number*、*email*、*url*、*search*、*tel* 和 *color*。适当的 *type* 声明是必需的，这样才能让 *input* 获得完整的样式。

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 输入框</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<form role="form">
  <div class="form-group">
    <label for="name">标签</label>
    <input type="text" class="form-control" placeholder="文本输入">
  </div>
 </form>

</body>
</html>
```

结果如下所示：

标签

文本输入

文本框（Textarea）

当您需要进行多行输入的时，则可以使用文本框 textarea。必要时可以改变 *rows* 属性（较少的行 = 较小的盒子，较多的行 = 较大的盒子）。

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 文本框</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<form role="form">
  <div class="form-group">
    <label for="name">文本框</label>
    <textarea class="form-control" rows="3"></textarea>
  </div>
</form>

</body>
</html>
```

结果如下所示：

文本框

复选框（CheckBoxe）和单选框（Radio）

复选框和单选按钮用于让用户从一系列预设置的选项中进行选择。

- 当创建表单时，如果您想让用户从列表中选择若干个选项时，请使用 *checkbox*。如果您限制用户只能选择一个选项，请使用 *radio*。
- 对一系列复选框和单选框使用 *.checkbox-inline* 或 *.radio-inline* class，控制它们显示在同一行上。

下面的实例演示了这两种类型（默认和内联）：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 复选框和单选按钮</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<label for="name">默认的复选框和单选按钮的实例</label>
<div class="checkbox">
   <label><input type="checkbox" value="">选项 1</label>
</div>
<div class="checkbox">
   <label><input type="checkbox" value="">选项 2</label>
</div>

<div class="radio">
   <label>
      <input type="radio" name="optionsRadios" id="optionsRadios1"
         value="option1" checked> 选项 1
   </label>
</div>
<div class="radio">
   <label>
      <input type="radio" name="optionsRadios" id="optionsRadios2"
         value="option2">
         选项 2 - 选择它将会取消选择选项 1
   </label>
</div>
<label for="name">内联的复选框和单选按钮的实例</label>
<div>
   <label class="checkbox-inline">
      <input type="checkbox" id="inlineCheckbox1" value="option1">
   </label>
   <label class="checkbox-inline">
      <input type="checkbox" id="inlineCheckbox2" value="option2">
   </label>
   <label class="checkbox-inline">
      <input type="checkbox" id="inlineCheckbox3" value="option3">
   </label>
   <label class="checkbox-inline">
      <input type="radio" name="optionsRadiosinline" id="optionsRad
         value="option1" checked> 选项 1
   </label>
   <label class="checkbox-inline">
      <input type="radio" name="optionsRadiosinline" id="optionsRad
         value="option2"> 选项 2
   </label>
</div>
```

```
</body>
</html>
```

结果如下所示：

默认的复选框和单选按钮的实例

☐ 选项 1
☐ 选项 2

◉ 选项 1
○ 选项 2 - 选择它将会取消选择选项 1

内联的复选框和单选按钮的实例

☐ 选项 1 ☐ 选项 2 ☐ 选项 3 ◉ 选项 1 ○ 选项 2

选择框（Select）

当您想让用户从多个选项中进行选择，但是默认情况下只能选择一个选项时，则使用选择框。

- 使用 <select> 展示列表选项，通常是那些用户很熟悉的选择列表，比如州或者数字。
- 使用 *multiple="multiple"* 允许用户选择多个选项。

下面的实例演示了这两种类型（select 和 multiple）：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 选择框</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<form role="form">
   <div class="form-group">
      <label for="name">选择列表</label>
      <select class="form-control">
         <option>1</option>
         <option>2</option>
         <option>3</option>
         <option>4</option>
         <option>5</option>
      </select>

      <label for="name">可多选的选择列表</label>
      <select multiple class="form-control">
         <option>1</option>
         <option>2</option>
         <option>3</option>
         <option>4</option>
         <option>5</option>
      </select>
   </div>
</form>

</body>
</html>
```

结果如下所示：

选择列表

1

可多选的选择列表

1
2
3
4

静态控件

当您需要在一个水平表单内的表单标签后放置纯文本时，请在 <p> 上使用 class *.form-control-static*。

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 静态控件</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<form class="form-horizontal" role="form">
  <div class="form-group">
    <label class="col-sm-2 control-label">Email</label>
    <div class="col-sm-10">
      <p class="form-control-static">email@example.com</p>
    </div>
  </div>
  <div class="form-group">
    <label for="inputPassword" class="col-sm-2 control-label">密码<
    <div class="col-sm-10">
      <input type="password" class="form-control" id="inputPassword
         placeholder="请输入密码">
    </div>
  </div>
</form>

</body>
</html>
```

结果如下所示：

Email email@example.com

密码 请输入密码

表单控件状态

除了 *:focus* 状态（即，用户点击 input 或使用 tab 键聚焦到 input 上），Bootstrap 还为禁用的输入框定义了样式，并提供了表单验证的 class。

输入框焦点

当输入框 input 接收到 *:focus* 时，输入框的轮廓会被移除，同时应用 *box-shadow*。

禁用的输入框 input

如果您想要禁用一个输入框 input，只需要简单地添加 *disabled* 属性，这不仅会禁用输入框，还会改变输入框的样式以及当鼠标的指针悬停在元素上时鼠标指针的样式。

禁用的字段集 fieldset

对 <fieldset> 添加 disabled 属性来禁用 <fieldset> 内的所有控件。

验证状态

Bootstrap 包含了错误、警告和成功消息的验证样式。只需要对父元素简单地添加适当的 class（*.has-warning*、 *.has-error* 或 *.has-success*）即可使用验证状态。

下面的实例演示了所有控件状态：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 表单控件状态</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<form class="form-horizontal" role="form">
   <div class="form-group">
      <label class="col-sm-2 control-label">聚焦</label>
      <div class="col-sm-10">
         <input class="form-control" id="focusedInput" type="text"
            value="该输入框获得焦点...">
      </div>
   </div>
   <div class="form-group">
      <label for="inputPassword" class="col-sm-2 control-label">
         禁用
      </label>
      <div class="col-sm-10">
         <input class="form-control" id="disabledInput" type="text"
            placeholder="该输入框禁止输入..." disabled>
      </div>
   </div>
   <fieldset disabled>
      <div class="form-group">
```

```
            <label for="disabledTextInput"  class="col-sm-2 control-la
               禁用输入（Fieldset disabled）
            </label>
            <div class="col-sm-10">
               <input type="text" id="disabledTextInput" class="form-c
                  placeholder="禁止输入">
            </div>
         </div>
         <div class="form-group">
            <label for="disabledSelect"  class="col-sm-2 control-label
               禁用选择菜单（Fieldset disabled）
            </label>
            <div class="col-sm-10">
               <select id="disabledSelect" class="form-control">
                  <option>禁止选择</option>
               </select>
            </div>
         </div>
      </fieldset>
      <div class="form-group has-success">
         <label class="col-sm-2 control-label" for="inputSuccess">
            输入成功
         </label>
         <div class="col-sm-10">
            <input type="text" class="form-control" id="inputSuccess">
         </div>
      </div>
      <div class="form-group has-warning">
         <label class="col-sm-2 control-label" for="inputWarning">
            输入警告
         </label>
         <div class="col-sm-10">
            <input type="text" class="form-control" id="inputWarning">
         </div>
      </div>
      <div class="form-group has-error">
         <label class="col-sm-2 control-label" for="inputError">
            输入错误
         </label>
         <div class="col-sm-10">
            <input type="text" class="form-control" id="inputError">
         </div>
      </div>
   </form>

</body>
</html>
```

结果如下所示：

聚焦 该输入框获得焦点...

禁用 该输入框禁止输入...

禁用输入（Fieldset disabled） 禁止输入

禁用选择菜单（Fieldset disabled） 禁止选择

输入成功

输入警告

输入错误

表单控件大小

您可以分别使用 class *.input-lg* 和 *.col-lg-** 来设置表单的高度和宽度。下面的实例演示了这点：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 表单控件大小</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<form role="form">
   <div class="form-group">
      <input class="form-control input-lg" type="text"
         placeholder=".input-lg">
   </div>

   <div class="form-group">
      <input class="form-control" type="text" placeholder="默认输入"
   </div>

   <div class="form-group">
      <input class="form-control input-sm" type="text"
         placeholder=".input-sm">
   </div>
   <div class="form-group">
   </div>
```

```
    <div class="form-group">
       <select class="form-control input-lg">
          <option value="">.input-lg</option>
       </select>
    </div>
    <div class="form-group">
       <select class="form-control">
          <option value="">默认选择</option>
       </select>
    </div>
    <div class="form-group">
       <select class="form-control input-sm">
          <option value="">.input-sm</option>
       </select>
    </div>

    <div class="row">
       <div class="col-lg-2">
          <input type="text" class="form-control" placeholder=".col-
       </div>
       <div class="col-lg-3">
          <input type="text" class="form-control" placeholder=".col-
       </div>
       <div class="col-lg-4">
          <input type="text" class="form-control" placeholder=".col-
       </div>
    </div>
 </form>

 </body>
 </html>
```

结果如下所示：

.input-lg

默认输入

.input-sm

.input-lg

默认选择

.input-sm

.col-lg-2

.col-lg-3

.col-lg-4

表单帮助文本

Bootstrap 表单控件可以在输入框 input 上有一个块级帮助文本。为了添加一个占用整个宽度的内容块，请在 <input> 后使用 *.help-block*。下面的实例演示了这点：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 表单帮助文本</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<form role="form">
   <span>帮助文本实例</span>
   <input class="form-control" type="text" placeholder="">
   <span class="help-block">一个较长的帮助文本块，超过一行，
   需要扩展到下一行。本实例中的帮助文本总共有两行。</span>
</form>

</body>
</html>
```

结果如下所示：

帮助文本实例

一个较长的帮助文本块，超过一行， 需要扩展到下一行。本实例中的帮助文本总共有两行。

Bootstrap 按钮

本章将通过实例讲解如何使用 Bootstrap 按钮。任何带有 class **.btn** 的元素都会继承圆角灰色按钮的默认外观。但是 Bootstrap 提供了一些选项来定义按钮的样式，具体如下表所示：

Class	描述
btn	默认的/标准的按钮。
btn-primary	提供额外的视觉效果，标识一组按钮中的原始动作。
btn-success	表示一个成功的或积极的动作。
btn-info	信息警告消息的上下文按钮。
btn-warning	表示应谨慎采取的动作。
btn-danger	表示一个危险的或潜在的负面动作。
btn-link	并不强调是一个按钮，看起来像一个链接，但同时保持按钮的行为。

下面的实例演示了上面所有的按钮 class：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 按钮选项</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<!-- 标准的按钮 -->
<button type="button" class="btn btn-default">默认按钮</button>

<!-- 提供额外的视觉效果，标识一组按钮中的原始动作 -->
<button type="button" class="btn btn-primary">原始按钮</button>

<!-- 表示一个成功的或积极的动作 -->
<button type="button" class="btn btn-success">成功按钮</button>

<!-- 信息警告消息的上下文按钮 -->
<button type="button" class="btn btn-info">信息按钮</button>

<!-- 表示应谨慎采取的动作 -->
<button type="button" class="btn btn-warning">警告按钮</button>

<!-- 表示一个危险的或潜在的负面动作 -->
<button type="button" class="btn btn-danger">危险按钮</button>

<!-- 并不强调是一个按钮，看起来像一个链接，但同时保持按钮的行为 -->
<button type="button" class="btn btn-link">链接按钮</button>

</body>
</html>
```

结果如下所示：



按钮大小

下表列出了获得各种大小按钮的 class：

Class	描述
.btn-lg	这会让按钮看起来比较大。
.btn-sm	这会让按钮看起来比较小。
.btn-xs	这会让按钮看起来特别小。
.btn-block	这会创建块级的按钮，会横跨父元素的全部宽度。

下面的实例演示了上面所有的按钮 class：

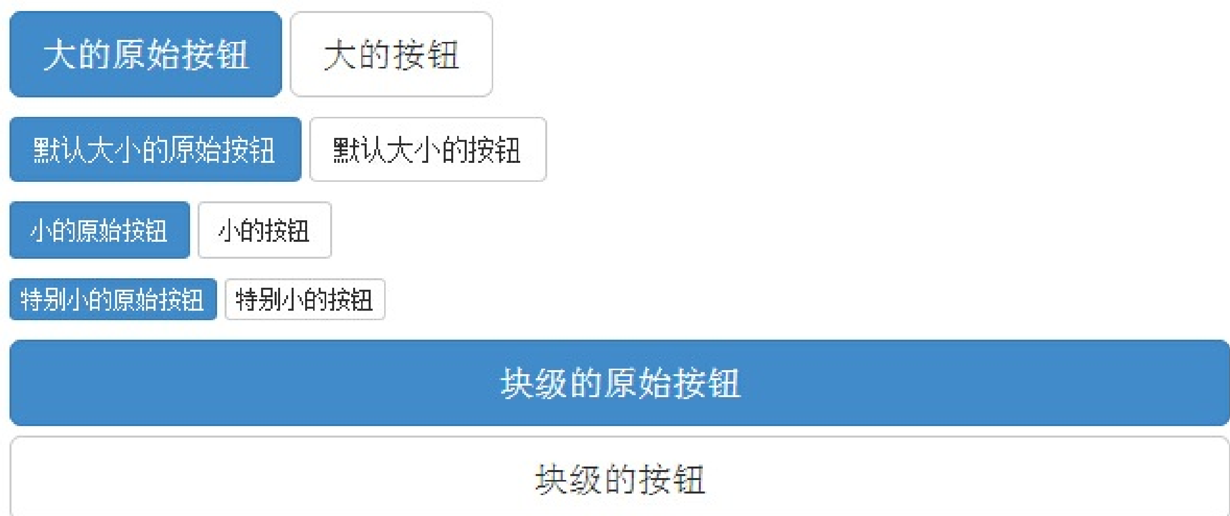
```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 按钮大小</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<p>
   <button type="button" class="btn btn-primary btn-lg">
      大的原始按钮
   </button>
   <button type="button" class="btn btn-default btn-lg">
      大的按钮
   </button>
</p>
<p>
   <button type="button" class="btn btn-primary">
      默认大小的原始按钮
   </button>
   <button type="button" class="btn btn-default">
      默认大小的按钮
   </button>
</p>
<p>
   <button type="button" class="btn btn-primary btn-sm">
      小的原始按钮
   </button>
   <button type="button" class="btn btn-default btn-sm">
      小的按钮
   </button>
</p>
<p>
   <button type="button" class="btn btn-primary btn-xs">
      特别小的原始按钮
   </button>
   <button type="button" class="btn btn-default btn-xs">
      特别小的按钮
```

```
    </button>
</p>
<p>
    <button type="button" class="btn btn-primary btn-lg btn-block">
        块级的原始按钮
    </button>
    <button type="button" class="btn btn-default btn-lg btn-block">
        块级的按钮
    </button>
</p>

</body>
</html>
```

结果如下所示：



按钮状态

Bootstrap 提供了激活、禁用等按钮状态的 class，下面将进行详细讲解。

激活状态

按钮在激活时将呈现为被按压的外观（深色的背景、深色的边框、阴影）。

下表列出了让按钮元素和锚元素呈激活状态的 class：

元素	Class
按钮元素	添加 **.active** class 来显示它是激活的。
锚元素	添加 **.active** class 到 <a> 按钮来显示它是激活的。

下面的实例演示了这点：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 按钮激活状态</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<p>
   <button type="button" class="btn btn-default btn-lg ">
      默认按钮
   </button>
   <button type="button" class="btn btn-default btn-lg active">
      激活按钮
   </button>
</p>
<p>
   <button type="button" class="btn btn-primary btn-lg ">
      原始按钮
   </button>
   <button type="button" class="btn btn-primary btn-lg active">
      激活的原始按钮
   </button>
</p>

</body>
</html>
```

结果如下所示：



禁用状态

当您禁用一个按钮时，它的颜色会变淡 50%，并失去渐变。

下表列出了让按钮元素和锚元素呈禁用状态的 class：

元素	Class
按钮元素	添加 **disabled** 属性 到 <button> 按钮。
锚元素	添加 **disabled** *class* 到 <a> 按钮。 注意：该 *class* 只会改变 <*a*> 的外观，不会改变它的功能。在这里，您需要使用自定义的 *JavaScript* 来禁用链接。

下面的实例演示了这点：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 按钮禁用状态</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<p>
   <button type="button" class="btn btn-default btn-lg">
      默认按钮
   </button>
   <button type="button" class="btn btn-default btn-lg" disabled="c
      禁用按钮
   </button>
</p>
<p>
   <button type="button" class="btn btn-primary btn-lg ">
      原始按钮
   </button>
   <button type="button" class="btn btn-primary btn-lg" disabled="c
      禁用的原始按钮
   </button>
</p>
<p>
   <a href="#" class="btn btn-default btn-lg" role="button">
      链接
   </a>
   <a href="#" class="btn btn-default btn-lg disabled" role="button
      禁用链接
   </a>
</p>
<p>
   <a href="#" class="btn btn-primary btn-lg" role="button">
      原始链接
   </a>
   <a href="#" class="btn btn-primary btn-lg disabled" role="button
      禁用的原始链接
   </a>
</p>

</body>
</html>
```

结果如下所示：



按钮标签

您可以在 <a>、<button> 或 <input> 元素上使用按钮 class。但是建议您在 <button> 元素上使用按钮 class，避免跨浏览器的不一致性问题。

下面的实例演示了这点：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 按钮标签</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<a class="btn btn-default" href="#" role="button">链接</a>
<button class="btn btn-default" type="submit">按钮</button>
<input class="btn btn-default" type="button" value="输入">
<input class="btn btn-default" type="submit" value="提交">

</body>
</html>
```

结果如下所示：



Bootstrap 图像

在本章中，我们将学习 Bootstrap 对图像的支持。Bootstrap 提供了三个可对图像应用简单样式的 class：

- *.img-rounded*：添加 *border-radius:6px* 来获得图像圆角。
- *.img-circle*：添加 *border-radius:500px* 来让整个图像变成圆形。
- *.img-thumbnail*：添加一些内边距（padding）和一个灰色的边框。

请看下面的实例演示：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 图像</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<img src="img/download.png"
   class="img-rounded">
<img src="img/download.png"
   class="img-circle">
<img src="img/download.png"
   class="img-thumbnail">

</body>
</html>
```

结果如下所示：



Bootstrap 帮助器类

本章将讨论 Bootstrap 中的一些可能会派上用场的帮助器类。

关闭图标

使用通用的关闭图标来关闭模态框和警告框。使用 class **close** 得到关闭图标。

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 关闭图标</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<p>关闭图标实例
   <button type="button" class="close" aria-hidden="true">
      &times;
   </button>
</p>

</body>
</html>
```

结果如下所示：

Close Icon Example ×

插入符

使用插入符表示下拉功能和方向。使用带有 class **caret** 的 <span> 元素得到该功能。

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 插入符</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<p>插入符实例
   <span class="caret"></span>
</p>

</body>
</html>
```

结果如下所示：

插入符实例 ▾

快速浮动

您可以分别使用 class **pull-left** 或 **pull-right** 来把元素向左或向右浮动。下面的实例演示了这点。

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 快速浮动</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="pull-left">
   向左快速浮动
</div>
<div class="pull-right">
   向右快速浮动
</div>

</body>
</html>
```

结果如下所示：

Quick Float to left Quick Float to right

如需对齐导航栏中的组件，请使用 **.navbar-left** 或 **.navbar-right** 代替。请查看 Bootstrap 导航栏。

居中内容块

使用 class **center-block** 来居中元素。

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 居中内容块</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="row">
   <div class="center-block" style="width:200px;background-color:#c
      这是 center-block 实例
   </div>
</div>

</body>
</html>
```

结果如下所示：

这是 center-block 实例

清除浮动

如需清除元素的浮动，请使用 **.clearfix** class。

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 清除浮动</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="clearfix"  style="background: #D8D8D8;border: 1px solid
   <div class="pull-left" style="background:#58D3F7;">
      向左快速浮动
   </div>
   <div class="pull-right" style="background: #DA81F5;">
      向右快速浮动
   </div>
</div>

</body>
</html>
```

结果如下所示：

向左快速浮动 向右快速浮动

显示和隐藏内容

您可以通过使用 class **.show** 和 **.hidden** 来强行设置元素显示或隐藏（包括屏幕阅读器）。

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 显示和隐藏内容</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="row" style="padding: 91px 100px 19px 50px;">
   <div class="show" style="left-margin:10px;width:300px;backgroun
      这是 show class 的实例
   </div>
   <div class="hidden" style="width:200px;background-color:#ccc;">
      这是 hide class 的实例
   </div>
</div>

</body>
</html>
```

结果如下所示：

This is an example for show class

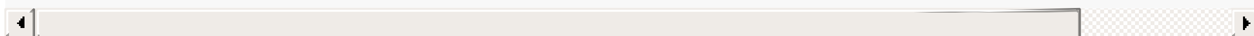
屏幕阅读器

您可以通过使用 class **.sr-only** 来把元素对所有设备隐藏，除了屏幕阅读器。

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 屏幕阅读器</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="row" style="padding: 91px 100px 19px 50px;">
   <form class="form-inline" role="form">
   <div class="form-group">
      <label class="sr-only" for="email">Email 地址</label>
      <input type="email" class="form-control" placeholder="Enter e
   </div>
   <div class="form-group">
      <label class="sr-only" for="pass">密码</label>
      <input type="password" class="form-control" placeholder="Pass
   </div>
</div>

</body>
</html>
```



结果如下所示：

Enter email　Password

在这里，我们看到两个 input 类型的 label 标签都带有 class **sr-only**，因此标签将只对屏幕阅读器可见。

Bootstrap 响应式实用工具

Bootstrap 提供了一些帮助器类，以便更快地实现对移动设备友好的开发。这些可以通过媒体查询结合大型、小型和中型设备，实现内容对设备的显示和隐藏。

需要谨慎使用这些工具，避免在同一个站点创建完全不同的版本。响应式实用工具目前只适用于块和表切换。

Class	设备
.visible-xs	额外的小设备（小于 768px）可见
.visible-sm	小型设备（768 px 起）可见
.visible-md	中型设备（768 px 到 991 px）可见
.visible-lg	大型设备（992 px 及以上）可见
.hidden-xs	额外的小设备（小于 768px）隐藏
.hidden-sm	小型设备（768 px 起）隐藏
.hidden-md	中型设备（768 px 到 991 px）隐藏
.hidden-lg	大型设备（992 px 及以上）隐藏

打印类

下表列出了打印类。使用这些切换打印内容。

Class	打印
.visible-print	可见，可打印
.hidden-print	只对浏览器可见，不可打印

实例

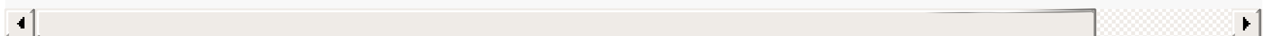
下面的实例演示了上面所列举的帮助器类的用法。调整浏览器的窗口大小，或者在不同的设备上加载实例，测试响应式实用工具类。

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 响应式实用工具</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="container" style="padding: 40px;">
   <div class="row visible-on">
      <div class="col-xs-6 col-sm-3" style="background-color: #dede
         box-shadow: inset 1px -1px 1px #444, inset -1px 1px 1px #4
         <span class="hidden-xs">特别小型</span>
         <span class="visible-xs">? 在特别小型设备上可见</span>
      </div>
      <div class="col-xs-6 col-sm-3" style="background-color: #dede
         box-shadow: inset 1px -1px 1px #444, inset -1px 1px 1px #4
         <span class="hidden-sm">小型</span>
         <span class="visible-sm">? 在小型设备上可见</span>
      </div>
      <div class="clearfix visible-xs"></div>
      <div class="col-xs-6 col-sm-3" style="background-color: #dede
         box-shadow: inset 1px -1px 1px #444, inset -1px 1px 1px #4
         <span class="hidden-md">中型</span>
         <span class="visible-md">? 在中型设备上可见</span>
      </div>
      <div class="col-xs-6 col-sm-3" style="background-color: #dede
         box-shadow: inset 1px -1px 1px #444, inset -1px 1px 1px #4
         <span class="hidden-lg">大型</span>
         <span class="visible-lg">? 在大型设备上可见</span>
      </div>
</div>

</body>
</html>
```

结果如下所示：



勾号（?） 表示元素在当前视口中可见。

Bootstrap 字形图标（Glyphicons）

本章将讲解字形图标（Glyphicons），并通过一些实例了解它的使用。Bootstrap 捆绑了 200 多种字体格式的自行。首先让我们先来理解一下什么是字形图标（Glyphicons）。

什么是字形图标（Glyphicons）？

字形图标（Glyphicons）是在 Web 项目中使用的图标字体。虽然，Glyphicons Halflings 需要商业许可，但是您可以通过基于项目的 Bootstrap 来免费使用这些图标。

为了表示对图标作者的感谢，希望您在使用时加上 GLYPHICONS 网站的链接。

获取字形图标（Glyphicons）

我们已经在 环境安装 章节下载了 Bootstrap 3.x 版本，并理解了它的目录结构。在 *fonts* 文件夹内可以找到字形图标（Glyphicons），它包含了下列这些文件：

- glyphicons-halflings-regular.eot
- glyphicons-halflings-regular.svg
- glyphicons-halflings-regular.ttf
- glyphicons-halflings-regular.woff

相关的 CSS 规则写在 *dist* 文件夹内的 css 文件夹内的 *bootstrap.css* 和 *bootstrap-min.css* 文件上。

字形图标（Glyphicons）列表

点击这里，查看可用的字形图标（Glyphicons）列表。

CSS 规则解释

下面的 CSS 规则构成 glyphicon class。

```
@font-face {
  font-family: 'Glyphicons Halflings';
  src: url('../fonts/glyphicons-halflings-regular.eot');
  src: url('../fonts/glyphicons-halflings-regular.eot?#iefix') forr
}

.glyphicon {
  position: relative;
  top: 1px;
  display: inline-block;
  font-family: 'Glyphicons Halflings';
  -webkit-font-smoothing: antialiased;
  font-style: normal;
  font-weight: normal;
  line-height: 1;
  -moz-osx-font-smoothing: grayscale;
}
```

所以 font-face 规则实际上是在找到 glyphicons 地方声明 font-family 和位置。

.glyphicon class 声明一个从顶部偏移 1px 的相对位置，呈现为 inline-block，声明字体，规定 font-style 和 font-weight 为 normal，设置行高为 1。除此之外，使用 *-webkit-font-smoothing: antialiased* 和 *-moz-osx-font-smoothing: grayscale;* 获得跨浏览器的一致性。

然后，这里的

```
.glyphicon:empty {
  width: 1em;
}
```

是空的 glyphicon。

这里有 200 个 class，每个 class 针对一个图标。这些 class 的常见格式如下：

```
.glyphicon-keyword:before {
  content: "hexvalue";
}
```

比如，使用的 user 图标，它的 class 如下：

```
.glyphicon-user:before {
  content: "\e008";
}
```

用法

如需使用图标，只需要简单地使用下面的代码即可。请在图标和文本之间保留适当的空间。

```
<span class="glyphicon glyphicon-search"></span>
```

下面的实例演示了如何使用字形图标（Glyphicons）：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 如何使用字形图标（Glyphicons）</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<p>
   <button type="button" class="btn btn-default">
      <span class="glyphicon glyphicon-sort-by-attributes"></span>
   </button>
   <button type="button" class="btn btn-default">
      <span class="glyphicon glyphicon-sort-by-attributes-alt"></sp
   </button>
   <button type="button" class="btn btn-default">
      <span class="glyphicon glyphicon-sort-by-order"></span>
   </button>
   <button type="button" class="btn btn-default">
      <span class="glyphicon glyphicon-sort-by-order-alt"></span>
   </button>
</p>
<button type="button" class="btn btn-default btn-lg">
  <span class="glyphicon glyphicon-user"></span> User
</button>
<button type="button" class="btn btn-default btn-sm">
  <span class="glyphicon glyphicon-user"></span> User
</button>
<button type="button" class="btn btn-default btn-xs">
  <span class="glyphicon glyphicon-user"></span> User
</button>

</body>
</html>
```

结果如下所示：



带有导航栏的字形图标（Glyphicons）

```
<!DOCTYPE html>
<html>
  <head>
    <title>导航栏的字形图标</title>
    <meta name="viewport" content="width=device-width, initial-scal
    <!-- Bootstrap -->
    <link href="http://apps.bdimg.com/libs/bootstrap/3.2.0/css/boot
    <style>
    body {
    padding-top: 50px;
    padding-left: 50px;
    }
    </style>
    <!--[if lt IE 9]>
      <script src="http://apps.bdimg.com/libs/html5shiv/3.7/html5sh
    <![endif]-->
  </head>
  <body>
    <div class="navbar navbar-fixed-top navbar-inverse" role="navig
      <div class="container">
        <div class="navbar-header">
          <button type="button" class="navbar-toggle" data-toggle="
            <span class="sr-only">Toggle navigation</span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
          </button>
          <a class="navbar-brand" href="#">Project name</a>
        </div>
        <div class="collapse navbar-collapse">
          <ul class="nav navbar-nav">
            <li class="active"><a href="#"><span class="glyphicon g
            <li><a href="#shop"><span class="glyphicon glyphicon-sh
            <li><a href="#support"><span class="glyphicon glyphicon
          </ul>
        </div><!-- /.nav-collapse -->
      </div><!-- /.container -->
    </div>
    <!-- jQuery (Bootstrap 插件需要引入) -->
    <script src="http://apps.bdimg.com/libs/jquery/2.1.1/jquery.min
    <!-- 包含了所有编译插件 -->
    <script src="http://apps.bdimg.com/libs/bootstrap/3.2.0/js/boot
  </body>
</html>
```

定制字形图标（Glyphicons）

我们已经看到如何使用字形图标（Glyphicons），接下来我们看看如何定制字形图标（Glyphicons）。

我们将以上面的实例开始，并通过改变字体尺寸、颜色和应用文本阴影来进行定制图标。

下面是开始的代码：

```
<button type="button" class="btn btn-primary btn-lg">
  <span class="glyphicon glyphicon-user"></span> User
</button>
```

效果如下所示：



定制字体尺寸

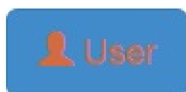
通过增加或减少图标的字体尺寸，您可以让图标看起来更大或更小。

```
<button type="button" class="btn btn-primary btn-lg" style="font-si
  <span class="glyphicon glyphicon-user"></span> User
</button>
```



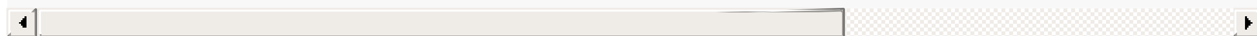
定制字体颜色

```
<button type="button" class="btn btn-primary btn-lg" style="color:
  <span class="glyphicon glyphicon-user"></span> User
</button>
```



应用文本阴影

```
<button type="button" class="btn btn-primary btn-lg" style="text-sh
  <span class="glyphicon glyphicon-user"></span> User
</button>
```





在线定制字形图标（Glyphicons）

点击这里，在线定制字形图标（Glyphicons）。

Bootstrap 下拉菜单（Dropdowns）

本章将重点介绍 Bootstrap 下拉菜单。下拉菜单是可切换的，是以列表格式显示链接的上下文菜单。这可以通过与 下拉菜单（Dropdown） JavaScript 插件 的互动来实现。

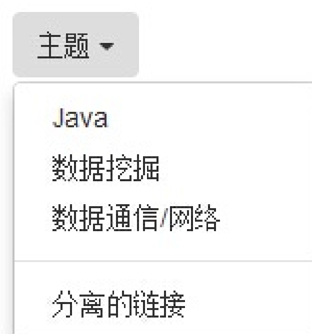
如需使用下列菜单，只需要在 class **.dropdown** 内加上下拉菜单即可。下面的实例演示了基本的下拉菜单：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 下拉菜单（Dropdowns）</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="dropdown">
   <button type="button" class="btn dropdown-toggle" id="dropdownMe
      data-toggle="dropdown">
      主题
      <span class="caret"></span>
   </button>
   <ul class="dropdown-menu" role="menu" aria-labelledby="dropdownM
      <li role="presentation">
         <a role="menuitem" tabindex="-1" href="#">Java</a>
      </li>
      <li role="presentation">
         <a role="menuitem" tabindex="-1" href="#">数据挖掘</a>
      </li>
      <li role="presentation">
         <a role="menuitem" tabindex="-1" href="#">
            数据通信/网络
         </a>
      </li>
      <li role="presentation" class="divider"></li>
      <li role="presentation">
         <a role="menuitem" tabindex="-1" href="#">分离的链接</a>
      </li>
   </ul>
</div>

</body>
</html>
```

结果如下所示：



选项

对齐

通过向 **.dropdown-menu** 添加 class **.pull-right** 来向右对齐下拉菜单。下面的实例演示了这点：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 向右对齐下拉菜单</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="dropdown">
   <button type="button" class="btn dropdown-toggle" id="dropdownMe
      data-toggle="dropdown">主题
      <span class="caret"></span>
   </button>
   <ul class="dropdown-menu pull-right" role="menu"
      aria-labelledby="dropdownMenu1">
      <li role="presentation">
         <a role="menuitem" tabindex="-1" href="#">Java</a>
      </li>
      <li role="presentation">
         <a role="menuitem" tabindex="-1" href="#">数据挖掘</a>
      </li>
      <li role="presentation">
         <a role="menuitem" tabindex="-1" href="#">
            数据通信/网络
         </a>
      </li>
      <li role="presentation" class="divider"></li>
      <li role="presentation">
         <a role="menuitem" tabindex="-1" href="#">分离的链接</a>
      </li>
   </ul>
</div>

</body>
</html>
```

结果如下所示：



标题

您可以使用 class **dropdown-header** 向下拉菜单的标签区域添加标题。下面的实例演示了这点：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 下拉菜单标题</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="dropdown">
   <button type="button" class="btn dropdown-toggle" id="dropdownMe
      data-toggle="dropdown">
      主题
      <span class="caret"></span>
   </button>
   <ul class="dropdown-menu" role="menu" aria-labelledby="dropdownM
      <li role="presentation" class="dropdown-header">下拉菜单标题</l
      <li role="presentation" >
         <a role="menuitem" tabindex="-1" href="#">Java</a>
      </li>
      <li role="presentation">
         <a role="menuitem" tabindex="-1" href="#">数据挖掘</a>
      </li>
      <li role="presentation">
         <a role="menuitem" tabindex="-1" href="#">
            数据通信/网络
         </a>
      </li>
      <li role="presentation" class="divider"></li>
      <li role="presentation" class="dropdown-header">下拉菜单标题</l
      <li role="presentation">
         <a role="menuitem" tabindex="-1" href="#">分离的链接</a>
      </li>
   </ul>
</div>

</body>
</html>
```

结果如下所示：

主题 ▾

下拉菜单标题
Java
数据挖掘
数据通信/网络

下拉菜单标题
分离的链接

Bootstrap 按钮组

按钮组允许多个按钮被堆叠在同一行上。当你想要把按钮对齐在一起时，这就显得非常有用。您可以通过 Bootstrap 按钮（Button） 插件 添加可选的 JavaScript 单选框和复选框样式行为。

下面的表格总结了 Bootstrap 提供的使用按钮组的一些重要的 class：

.btn-group

该 class 用于形成基本的按钮组。在 **.btn-group** 中放置一系列带有 class **.btn** 的按钮。

```
&lt;div class="btn-group"&gt;
  &lt;button type="button" class="btn btn-default"&gt;Button1&lt;/b
   &lt;button type="button" class="btn btn-default"&gt;Button2&lt;/
&lt;/div&gt;
```

.btn-toolbar

该 class 有助于把几组 <div class="btn-group"> 结合到一个 <div class="btn-toolbar"> 中，一般获得更复杂的组件。

```
&lt;div class="btn-toolbar" role="toolbar"&gt;
  &lt;div class="btn-group"&gt;...&lt;/div&gt;
  &lt;div class="btn-group"&gt;...&lt;/div&gt;
&lt;/div&gt;
```

.btn-group-lg, .btn-group-sm, .btn-group-xs

这些 class 可应用到整个按钮组的大小调整，而不需要对每个按钮进行大小调整。

```
&lt;div class="btn-group btn-group-lg"&gt;...&lt;/div&gt;
&lt;div class="btn-group btn-group-sm"&gt;...&lt;/div&gt;
&lt;div class="btn-group btn-group-xs"&gt;...&lt;/div&gt;
```

.btn-group-vertical

该 class 让一组按钮垂直堆叠显示，而不是水平堆叠显示。

```
&lt;div class="btn-group-vertical"&gt;
  ...
&lt;/div&gt;
```

基本的按钮组

下面的实例演示了上面表格中讨论到的 class **.btn-group** 的使用：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 基本的按钮组</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="btn-group">
  <button type="button" class="btn btn-default">按钮 1</button>
  <button type="button" class="btn btn-default">按钮 2</button>
  <button type="button" class="btn btn-default">按钮 3</button>
</div>

</body>
</html>
```

结果如下所示：



按钮工具栏

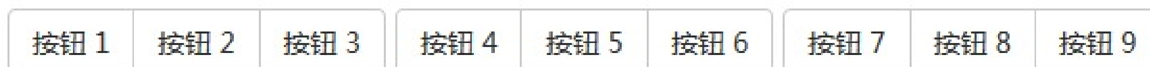
下面的实例演示了上面表格中讨论到的 class **.btn-toolbar** 的使用：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 按钮工具栏</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="btn-toolbar" role="toolbar">
  <div class="btn-group">
  <button type="button" class="btn btn-default">按钮 1</button>
  <button type="button" class="btn btn-default">按钮 2</button>
  <button type="button" class="btn btn-default">按钮 3</button>
 </div>
  <div class="btn-group">
  <button type="button" class="btn btn-default">按钮 4</button>
  <button type="button" class="btn btn-default">按钮 5</button>
  <button type="button" class="btn btn-default">按钮 6</button>
  </div>
  <div class="btn-group">
  <button type="button" class="btn btn-default">按钮 7</button>
  <button type="button" class="btn btn-default">按钮 8</button>
  <button type="button" class="btn btn-default">按钮 9</button>
  </div>
</div>

</body>
</html>
```

结果如下所示：



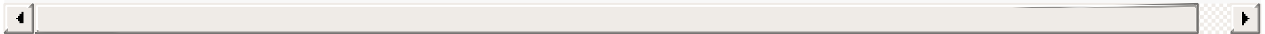
按钮的大小

下面的实例演示了上面表格中讨论到的 class **.btn-group-*** 的使用：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 按钮组的大小</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="btn-group btn-group-lg">
  <button type="button" class="btn btn-default">按钮 1</button>
  <button type="button" class="btn btn-default">按钮 2</button>
  <button type="button" class="btn btn-default">按钮 3</button>
 </div>
  <div class="btn-group btn-group-sm">
  <button type="button" class="btn btn-default">按钮 4</button>
  <button type="button" class="btn btn-default">按钮 5</button>
  <button type="button" class="btn btn-default">按钮 6</button>
</div>
  <div class="btn-group btn-group-xs">
  <button type="button" class="btn btn-default">按钮 7</button>
  <button type="button" class="btn btn-default">按钮 8</button>
  <button type="button" class="btn btn-default">按钮 9</button>
</div>

</body>
</html>
```



结果如下所示：



嵌套

您可以在一个按钮组内嵌套另一个按钮组，即，在一个 **.btn-group** 内嵌套另一个 **.btn-group** 。当您向下拉菜单与一系列按钮组合使用时，就会用到这个。

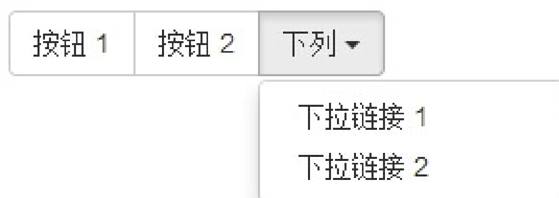
```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 嵌套的按钮组</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="btn-group">
  <button type="button" class="btn btn-default">按钮 1</button>
  <button type="button" class="btn btn-default">按钮 2</button>

  <div class="btn-group">
    <button type="button" class="btn btn-default dropdown-toggle"
      data-toggle="dropdown">
      下列
      <span class="caret"></span>
    </button>
    <ul class="dropdown-menu">
      <li><a href="#">下拉链接 1</a></li>
      <li><a href="#">下拉链接 2</a></li>
    </ul>
  </div>
</div>

</body>
</html>
```

结果如下所示：



垂直的按钮组

下面的实例演示了上面表格中讨论到的 class **.btn-group-vertical** 的使用：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 垂直的按钮组</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="btn-group-vertical">
  <button type="button" class="btn btn-default">按钮 1</button>
  <button type="button" class="btn btn-default">按钮 2</button>

  <div class="btn-group-vertical">
    <button type="button" class="btn btn-default dropdown-toggle"
      data-toggle="dropdown">
      下拉
      <span class="caret"></span>
    </button>
    <ul class="dropdown-menu">
      <li><a href="#">下拉链接 1</a></li>
      <li><a href="#">下拉链接 2</a></li>
    </ul>
  </div>
</div>

</body>
</html>
```

结果如下所示：



Bootstrap 按钮下拉菜单

本章将讲解如何使用 Bootstrap class 向按钮添加下拉菜单。如需向按钮添加下拉菜单，只需要简单地在在一个 **.btn-group** 中放置按钮和下拉菜单即可。您也可以使用 来指示按钮作为下拉菜单。

下面的实例演示了一个基本的简单的按钮下拉菜单：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 基本的按钮下拉菜单</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="btn-group">
   <button type="button" class="btn btn-default dropdown-toggle"
      data-toggle="dropdown">
      默认 <span class="caret"></span>
   </button>
   <ul class="dropdown-menu" role="menu">
      <li><a href="#">功能</a></li>
      <li><a href="#">另一个功能</a></li>
      <li><a href="#">其他</a></li>
      <li class="divider"></li>
      <li><a href="#">分离的链接</a></li>
   </ul>
</div>
<div class="btn-group">
   <button type="button" class="btn btn-primary dropdown-toggle"
      data-toggle="dropdown">
      原始 <span class="caret"></span>
   </button>
   <ul class="dropdown-menu" role="menu">
      <li><a href="#">功能</a></li>
      <li><a href="#">另一个功能</a></li>
      <li><a href="#">其他</a></li>
      <li class="divider"></li>
      <li><a href="#">分离的链接</a></li>
   </ul>
</div>

</body>
</html>
```

结果如下所示：



分割的按钮下拉菜单

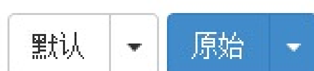
分割的按钮下拉菜单使用与下拉菜单按钮大致相同的样式，但是对下拉菜单添加了原始的功能。分割按钮的左边是原始的功能，右边是显示下拉菜单的切换。

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 分割的按钮下拉菜单</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="btn-group">
   <button type="button" class="btn btn-default">默认</button>
   <button type="button" class="btn btn-default dropdown-toggle"
      data-toggle="dropdown">
      <span class="caret"></span>
      <span class="sr-only">切换下拉菜单</span>
   </button>
   <ul class="dropdown-menu" role="menu">
      <li><a href="#">功能</a></li>
      <li><a href="#">另一个功能</a></li>
      <li><a href="#">其他</a></li>
      <li class="divider"></li>
      <li><a href="#">分离的链接</a></li>
   </ul>
</div>
<div class="btn-group">
   <button type="button" class="btn btn-primary">原始</button>
   <button type="button" class="btn btn-primary dropdown-toggle"
      data-toggle="dropdown">
      <span class="caret"></span>
      <span class="sr-only">切换下拉菜单</span>
   </button>
   <ul class="dropdown-menu" role="menu">
      <li><a href="#">功能</a></li>
      <li><a href="#">另一个功能</a></li>
      <li><a href="#">其他</a></li>
      <li class="divider"></li>
      <li><a href="#">分离的链接</a></li>
   </ul>
</div>

</body>
</html>
```

结果如下所示：



按钮下拉菜单的大小

您可以使用带有各种大小按钮的下拉菜单：**.btn-large**、**.btn-sm** 或 **.btn-xs**。

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 按钮下拉菜单的大小</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="btn-group">
   <button type="button" class="btn btn-default dropdown-toggle btn
      data-toggle="dropdown">
      默认 <span class="caret"></span>
   </button>
   <ul class="dropdown-menu" role="menu">
      <li><a href="#">功能</a></li>
      <li><a href="#">另一个功能</a></li>
      <li><a href="#">其他</a></li>
      <li class="divider"></li>
      <li><a href="#">分离的链接</a></li>
   </ul>
</div>
<div class="btn-group">
   <button type="button" class="btn btn-primary dropdown-toggle btn
      data-toggle="dropdown">
      原始 <span class="caret"></span>
   </button>
   <ul class="dropdown-menu" role="menu">
      <li><a href="#">功能</a></li>
      <li><a href="#">另一个功能</a></li>
      <li><a href="#">其他</a></li>
      <li class="divider"></li>
      <li><a href="#">分离的链接</a></li>
   </ul>
</div>
<div class="btn-group">
   <button type="button" class="btn btn-success dropdown-toggle btn
      data-toggle="dropdown">
      成功 <span class="caret"></span>
   </button>
   <ul class="dropdown-menu" role="menu">
      <li><a href="#">功能</a></li>
      <li><a href="#">另一个功能</a></li>
      <li><a href="#">其他</a></li>
      <li class="divider"></li>
      <li><a href="#">分离的链接</a></li>
```

```
    </ul>
</div>

</body>
</html>
```

结果如下所示：



按钮上拉菜单

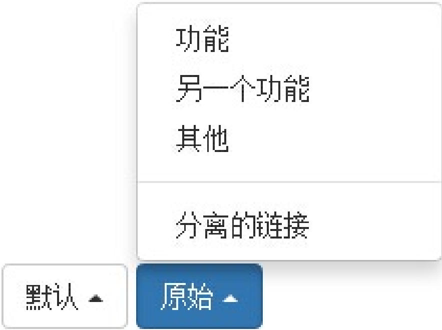
菜单也可以往上拉伸的，只需要简单地向父 **.btn-group** 容器添加 **.dropup** 即可。

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 按钮上拉菜单</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="row" style="margin-left:50px; margin-top:200px">
   <div class="btn-group dropup">
      <button type="button" class="btn btn-default dropdown-toggle'
         data-toggle="dropdown">
         默认 <span class="caret"></span>
      </button>
      <ul class="dropdown-menu" role="menu">
         <li><a href="#">功能</a></li>
         <li><a href="#">另一个功能</a></li>
         <li><a href="#">其他</a></li>
         <li class="divider"></li>
         <li><a href="#">分离的链接</a></li>
      </ul>
   </div>
   <div class="btn-group dropup">
      <button type="button" class="btn btn-primary dropdown-toggle'
         data-toggle="dropdown">
         原始 <span class="caret"></span>
      </button>
      <ul class="dropdown-menu" role="menu">
         <li><a href="#">功能</a></li>
         <li><a href="#">另一个功能</a></li>
         <li><a href="#">其他</a></li>
         <li class="divider"></li>
         <li><a href="#">分离的链接</a></li>
      </ul>
   </div>
</div>

</body>
</html>
```

结果如下所示：


功能
另一个功能
其他
分离的链接
默认
原始

Bootstrap 输入框组

本章将讲解 Bootstrap 支持的另一个特性，输入框组。输入框组扩展自 表单控件。使用输入框组，您可以很容易地向基于文本的输入框添加作为前缀和后缀的文本或按钮。

通过向输入域添加前缀和后缀的内容，您可以向用户输入添加公共的元素。例如，您可以添加美元符号，或者在 Twitter 用户名前添加 @，或者应用程序接口所需要的其他公共的元素。

向 **.form-control** 添加前缀或后缀元素的步骤如下：

- 把前缀后后缀元素放在一个带有 class **.input-group** 的 <div> 中。
- 接着，在相同的 <div> 内，在 class 为 **.input-group-addon** 的 <span> 内放置额外的内容。
- 把该 <span> 放置在 <input> 元素的前面或者后面。

> 为了保持跨浏览器的兼容性，请避免使用 <select> 元素，因为它们在 WebKit 浏览器中不能完全渲染出效果。也不要直接向表单组应用输入框组的 class，输入框组是一个孤立的组件。

基本的输入框组

下面的实例演示了基本的输入框组：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 基本的输入框组</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div style="padding: 100px 100px 10px;">
   <form class="bs-example bs-example-form" role="form">
      <div class="input-group">
         <span class="input-group-addon">@</span>
         <input type="text" class="form-control" placeholder="twitt
      </div>
      <br>

      <div class="input-group">
         <input type="text" class="form-control">
         <span class="input-group-addon">.00</span>
      </div>
      <br>
      <div class="input-group">
         <span class="input-group-addon">$</span>
         <input type="text" class="form-control">
         <span class="input-group-addon">.00</span>
      </div>
   </form>
</div>

</body>
</html>
```

结果如下所示：

@	twitterhandle

	.00

$		.00

输入框组的大小

您可以通过向 **.input-group** 添加相对表单大小的 class（比如 **.input-group-lg**、**input-group-sm**、**input-group-xs**）来改变输入框组的大小。输入框中的内容会自动调整大小。

下面的实例演示了这点：

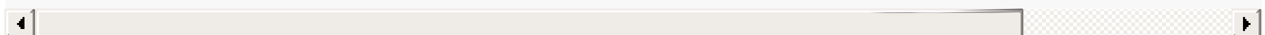
```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 输入框组的大小</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div style="padding: 100px 100px 10px;">
   <form class="bs-example bs-example-form" role="form">
      <div class="input-group input-group-lg">
         <span class="input-group-addon">@</span>
         <input type="text" class="form-control" placeholder="Twitt
      </div><br>

      <div class="input-group">
         <span class="input-group-addon">@</span>
         <input type="text" class="form-control" placeholder="Twitt
      </div><br>

      <div class="input-group input-group-sm">
         <span class="input-group-addon">@</span>
         <input type="text" class="form-control" placeholder="Twitt
      </div>
   </form>
</div>

</body>
</html>
```



结果如下所示：

@	Twitterhandle
@	Twitterhandle
@	Twitterhandle

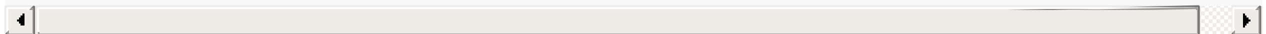
复选框和单选插件

您可以把复选框和单选插件作为输入框组的前缀或者后缀元素，如下面的实例所示：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 输入框组的复选框和单选插件</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div style="padding: 100px 100px 10px;">
   <form class="bs-example bs-example-form" role="form">
      <div class="row">
         <div class="col-lg-6">
            <div class="input-group">
               <span class="input-group-addon">
                  <input type="checkbox">
               </span>
               <input type="text" class="form-control">
            </div><!-- /input-group -->
         </div><!-- /.col-lg-6 --><br>
         <div class="col-lg-6">
            <div class="input-group">
               <span class="input-group-addon">
                  <input type="radio">
               </span>
               <input type="text" class="form-control">
            </div><!-- /input-group -->
         </div><!-- /.col-lg-6 -->
      </div><!-- /.row -->
   </form>
</div>

</body>
</html>
```



结果如下所示：

按钮插件

您也可以把按钮作为输入框组的前缀或者后缀元素，这个时候您就不是添加 **.input-group-addon** class，您需要使用 class **.input-group-btn** 来包裹按钮。这是必需的，因为默认的浏览器样式不会被重写。下面的实例演示了这点：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 输入框组的按钮插件</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div style="padding: 100px 100px 10px;">
   <form class="bs-example bs-example-form" role="form">
      <div class="row">
         <div class="col-lg-6">
            <div class="input-group">
               <span class="input-group-btn">
                  <button class="btn btn-default" type="button">
                     Go!
                  </button>
               </span>
               <input type="text" class="form-control">
            </div><!-- /input-group -->
         </div><!-- /.col-lg-6 --><br>
         <div class="col-lg-6">
            <div class="input-group">
               <input type="text" class="form-control">
               <span class="input-group-btn">
                  <button class="btn btn-default" type="button">
                     Go!
                  </button>
               </span>
            </div><!-- /input-group -->
         </div><!-- /.col-lg-6 -->
      </div><!-- /.row -->
   </form>
</div>

</body>
</html>
```

结果如下所示：

Go!

Go!

带有下拉菜单的按钮

在输入框组中添加带有下拉菜单的按钮，只需要简单地在一个 **.input-group-btn** class 中包裹按钮和下拉菜单即可，如下面的实例所示：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 输入框组的下拉菜单按钮</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div style="padding: 100px 100px 10px;">
   <form class="bs-example bs-example-form" role="form">
      <div class="row">
         <div class="col-lg-6">
            <div class="input-group">
               <div class="input-group-btn">
                  <button type="button" class="btn btn-default
                     dropdown-toggle" data-toggle="dropdown">
                     下拉菜单
                     <span class="caret"></span>
                  </button>
                  <ul class="dropdown-menu">
                     <li><a href="#">功能</a></li>
                     <li><a href="#">另一个功能</a></li>
                     <li><a href="#">其他</a></li>
                     <li class="divider"></li>
                     <li><a href="#">分离的链接</a></li>
                  </ul>
               </div><!-- /btn-group -->
               <input type="text" class="form-control">
            </div><!-- /input-group -->
         </div><!-- /.col-lg-6 --><br>
         <div class="col-lg-6">
            <div class="input-group">
               <input type="text" class="form-control">
               <div class="input-group-btn">
                  <button type="button" class="btn btn-default
                     dropdown-toggle" data-toggle="dropdown">
                     下拉菜单
```

```
                     <span class="caret"></span>
                  </button>
                  <ul class="dropdown-menu pull-right">
                     <li><a href="#">功能</a></li>
                     <li><a href="#">另一个功能</a></li>
                     <li><a href="#">其他</a></li>
                     <li class="divider"></li>
                     <li><a href="#">分离的链接</a></li>
                  </ul>
               </div><!-- /btn-group -->
            </div><!-- /input-group -->
         </div><!-- /.col-lg-6 -->
      </div><!-- /.row -->
   </form>
</div>

</body>
</html>
```

结果如下所示：

下拉菜单 ▾

下拉菜单 ▾

分割的下拉菜单按钮

在输入框组中添加带有下拉菜单的分割按钮，使用与下拉菜单按钮大致相同的样式，但是对下拉菜单添加了主要的功能，如下面的实例所示：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 输入框组中分割的下拉菜单按钮</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div style="padding: 100px 100px 10px;">
   <form class="bs-example bs-example-form" role="form">
      <div class="row">
         <div class="col-lg-6">
            <div class="input-group">
               <div class="input-group-btn">
```

```
                    <button type="button" class="btn btn-default"
                       tabindex="-1">下拉菜单
                    </button>
                    <button type="button" class="btn btn-default
                       dropdown-toggle" data-toggle="dropdown" tabinc
                       <span class="caret"></span>
                       <span class="sr-only">切换下拉菜单</span>
                    </button>
                    <ul class="dropdown-menu">
                       <li><a href="#">功能</a></li>
                       <li><a href="#">另一个功能</a></li>
                       <li><a href="#">其他</a></li>
                       <li class="divider"></li>
                       <li><a href="#">分离的链接</a></li>
                    </ul>
                 </div><!-- /btn-group -->
                 <input type="text" class="form-control">
              </div><!-- /input-group -->
           </div><!-- /.col-lg-6 --><br>
           <div class="col-lg-6">
              <div class="input-group">
                 <input type="text" class="form-control">
                 <div class="input-group-btn">
                    <button type="button" class="btn btn-default"
                       tabindex="-1">下拉菜单
                    </button>
                    <button type="button" class="btn btn-default
                       dropdown-toggle" data-toggle="dropdown" tabinc
                       <span class="caret"></span>
                       <span class="sr-only">切换下拉菜单</span>
                    </button>
                    <ul class="dropdown-menu pull-right">
                       <li><a href="#">功能</a></li>
                       <li><a href="#">另一个功能</a></li>
                       <li><a href="#">其他</a></li>
                       <li class="divider"></li>
                       <li><a href="#">分离的链接</a></li>
                    </ul>
                 </div><!-- /btn-group -->
              </div><!-- /input-group -->
           </div><!-- /.col-lg-6 -->
        </div><!-- /.row -->
     </form>
  </div>

  </body>
  </html>
```

结果如下所示：

下拉菜单 ▾

下拉菜单 ▾

Bootstrap 导航元素

本章我们将讲解 Bootstrap 提供的用于定义导航元素的一些选项。它们使用相同的标记和基类 **.nav**。Bootstrap 也提供了一个用于共享标记和状态的帮助器类。改变修饰的 class，可以在不同的样式间进行切换。

表格导航或标签

创建一个标签式的导航菜单：

- 以一个带有 class **.nav** 的无序列表开始。
- 添加 class **.nav-tabs**。

下面的实例演示了这点：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 标签式的导航菜单</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<p>标签式的导航菜单</p>
<ul class="nav nav-tabs">
   <li class="active"><a href="#">Home</a></li>
   <li><a href="#">SVN</a></li>
   <li><a href="#">iOS</a></li>
   <li><a href="#">VB.Net</a></li>
   <li><a href="#">Java</a></li>
   <li><a href="#">PHP</a></li>
</ul>

</body>
</html>
```

结果如下所示：

标签式的导航菜单



胶囊式的导航菜单

基本的胶囊式导航菜单

如果需要把标签改成胶囊的样式，只需要使用 class **.nav-pills** 代替 **.nav-tabs** 即可，其他的步骤与上面相同。

下面的实例演示了这点：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 基本的胶囊式导航菜单</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<p>基本的胶囊式导航菜单</p>
<ul class="nav nav-pills">
   <li class="active"><a href="#">Home</a></li>
   <li><a href="#">SVN</a></li>
   <li><a href="#">iOS</a></li>
   <li><a href="#">VB.Net</a></li>
   <li><a href="#">Java</a></li>
   <li><a href="#">PHP</a></li>
</ul>

</body>
</html>
```

结果如下所示：

基本的胶囊式导航菜单



垂直的胶囊式导航菜单

您可以在使用 class **.nav**、**.nav-pills** 的同时使用 class **.nav-stacked**，让胶囊垂直堆叠。

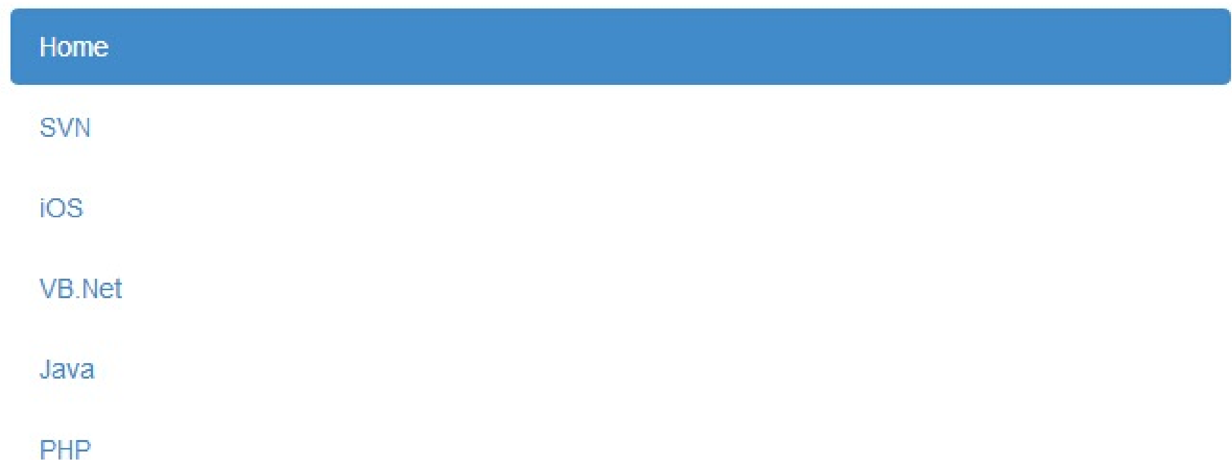
下面的实例演示了这点：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 垂直的胶囊式导航菜单</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<p>垂直的胶囊式导航菜单</p>
<ul class="nav nav-pills nav-stacked">
   <li class="active"><a href="#">Home</a></li>
   <li><a href="#">SVN</a></li>
   <li><a href="#">iOS</a></li>
   <li><a href="#">VB.Net</a></li>
   <li><a href="#">Java</a></li>
   <li><a href="#">PHP</a></li>
</ul>

</body>
</html>
```

结果如下所示：



两端对齐的导航

您可以在屏幕宽度大于 768px 时，通过在分别使用 **.nav**、**.nav-tabs** 或 **.nav**、**.nav-pills** 的同时使用 class **.nav-justified**，让标签式或胶囊式导航菜单与父元素等宽。在更小的屏幕上，导航链接会堆叠。

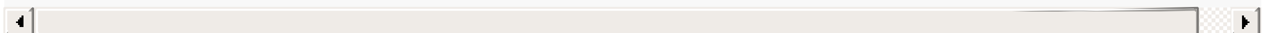
下面的实例演示了这点：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 两端对齐的导航元素</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<p>两端对齐的导航元素</p>
<ul class="nav nav-pills nav-justified">
   <li class="active"><a href="#">Home</a></li>
   <li><a href="#">SVN</a></li>
   <li><a href="#">iOS</a></li>
   <li><a href="#">VB.Net</a></li>
   <li><a href="#">Java</a></li>
   <li><a href="#">PHP</a></li>
</ul><br><br><br>

<ul class="nav nav-tabs nav-justified">
   <li class="active"><a href="#">Home</a></li>
   <li><a href="#">SVN</a></li>
   <li><a href="#">iOS</a></li>
   <li><a href="#">VB.Net</a></li>
   <li><a href="#">Java</a></li>
   <li><a href="#">PHP</a></li>
</ul>

</body>
</html>
```



结果如下所示：

两端对齐的导航元素



禁用链接

对每个 **.nav** class，如果添加了 **.disabled** class，则会创建一个灰色的链接，同时禁用了该链接的 **:hover** 状态，如下面的实例所示：

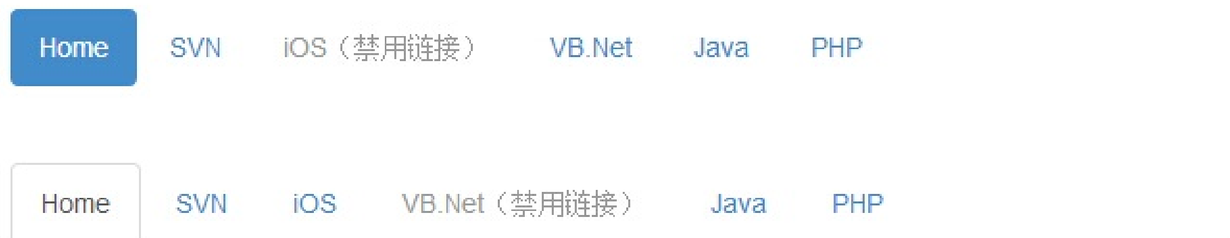
```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 导航元素中的禁用链接</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<p>导航元素中的禁用链接</p>
<ul class="nav nav-pills">
   <li class="active"><a href="#">Home</a></li>
   <li><a href="#">SVN</a></li>
   <li class="disabled"><a href="#">iOS（禁用链接）</a></li>
   <li><a href="#">VB.Net</a></li>
   <li><a href="#">Java</a></li>
   <li><a href="#">PHP</a></li>
</ul><br><br>

<ul class="nav nav-tabs">
   <li class="active"><a href="#">Home</a></li>
   <li><a href="#">SVN</a></li>
   <li><a href="#">iOS</a></li>
   <li  class="disabled"><a href="#">VB.Net（禁用链接）</a></li>
   <li><a href="#">Java</a></li>
   <li><a href="#">PHP</a></li>
</ul>

</body>
</html>
```

结果如下所示：



> 该 class 只会改变 <a> 的外观，不会改变它的功能。在这里，您需要使用自定义的 JavaScript 来禁用链接。

下拉菜单

导航菜单与下拉菜单使用相似的语法。默认情况下，列表项的锚与一些数据属性协同合作来触发带有 **.dropdown-menu** class 的无序列表。

带有下拉菜单的标签

向标签添加下拉菜单的步骤如下：

- 以一个带有 class **.nav** 的无序列表开始。
- 添加 class **.nav-tabs**。
- 添加带有 **.dropdown-menu** class 的无序列表。

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 带有下拉菜单的标签</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<p>带有下拉菜单的标签</p>
<ul class="nav nav-tabs">
   <li class="active"><a href="#">Home</a></li>
   <li><a href="#">SVN</a></li>
   <li><a href="#">iOS</a></li>
   <li><a href="#">VB.Net</a></li>
   <li class="dropdown">
      <a class="dropdown-toggle" data-toggle="dropdown" href="#">
         Java <span class="caret"></span>
      </a>
      <ul class="dropdown-menu">
         <li><a href="#">Swing</a></li>
         <li><a href="#">jMeter</a></li>
         <li><a href="#">EJB</a></li>
         <li class="divider"></li>
         <li><a href="#">分离的链接</a></li>
      </ul>
   </li>
   <li><a href="#">PHP</a></li>
</ul>

</body>
</html>
```

结果如下所示：



带有下拉菜单的胶囊

步骤与创建带有下拉菜单的标签相同，只是需要把 **.nav-tabs** class 改为 **.nav-pills**，如下面的实例所示：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 带有下拉菜单的胶囊</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<p>带有下拉菜单的胶囊</p>
<ul class="nav nav-pills">
   <li class="active"><a href="#">Home</a></li>
   <li><a href="#">SVN</a></li>
   <li><a href="#">iOS</a></li>
   <li><a href="#">VB.Net</a></li>
   <li class="dropdown">
      <a class="dropdown-toggle" data-toggle="dropdown" href="#">
         Java <span class="caret"></span>
      </a>
      <ul class="dropdown-menu">
         <li><a href="#">Swing</a></li>
         <li><a href="#">jMeter</a></li>
         <li><a href="#">EJB</a></li>
         <li class="divider"></li>
         <li><a href="#">分离的链接</a></li>
      </ul>
   </li>
   <li><a href="#">PHP</a></li>
</ul>

</body>
</html>
```

结果如下所示：



Bootstrap 导航栏

导航栏是一个很好的功能，是 Bootstrap 网站的一个突出特点。导航栏是响应式元组件就，作为应用程序或网站的导航标题。导航栏在移动设备的视图中是折叠的，随着可用视口宽度的增加，导航栏也会水平展开。在 Bootstrap 导航栏的核心中，导航栏包括了为站点名称和基本的导航定义样式。

默认的导航栏

创建一个默认的导航栏的步骤如下：

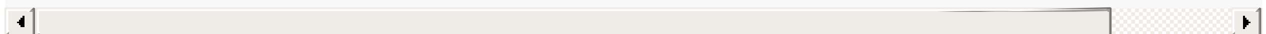
- 向 <nav> 标签添加 class **.navbar**、**.navbar-default**。
- 向上面的元素添加 **role="navigation"**，有助于增加可访问性。
- 向 <div> 元素添加一个标题 class **.navbar-header**，内部包含了带有 class **navbar-brand** 的 <a> 元素。这会让文本看起来更大一号。
- 为了向导航栏添加链接，只需要简单地添加带有 class **.nav**、**.navbar-nav** 的无序列表即可。

下面的实例演示了这点：

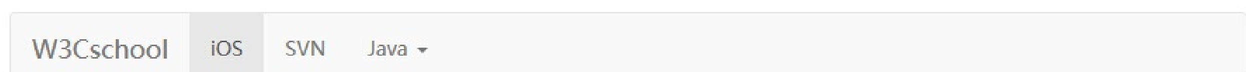
```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 默认的导航栏</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<nav class="navbar navbar-default" role="navigation">
   <div class="navbar-header">
      <a class="navbar-brand" href="#">W3Cschool</a>
   </div>
   <div>
      <ul class="nav navbar-nav">
         <li class="active"><a href="#">iOS</a></li>
         <li><a href="#">SVN</a></li>
         <li class="dropdown">
            <a href="#" class="dropdown-toggle" data-toggle="dropdo
               Java
               <b class="caret"></b>
            </a>
            <ul class="dropdown-menu">
               <li><a href="#">jmeter</a></li>
               <li><a href="#">EJB</a></li>
               <li><a href="#">Jasper Report</a></li>
               <li class="divider"></li>
               <li><a href="#">分离的链接</a></li>
               <li class="divider"></li>
               <li><a href="#">另一个分离的链接</a></li>
            </ul>
         </li>
      </ul>
   </div>
</nav>

</body>
</html>
```



结果如下所示：



响应式的导航栏

为了给导航栏添加响应式特性，您要折叠的内容必须包裹在带有 classes **.collapse**、**.navbar-collapse** 的 <div> 中。折叠起来的导航栏实际上是一个带有 class **.navbar-toggle** 及两个 data- 元素的按钮。第一个是 **data-toggle**，用于告诉 JavaScript 需要对按钮做什么，第二个是 **data-target**，指示要切换到哪一个元素。三个带有 class **.icon-bar** 的 <span> 创建所谓的汉堡按钮。这些会切换为 **.nav-collapse** <div> 中的元素。为了实现以上这些功能，您必须包含 Bootstrap 折叠（Collapse）插件。

下面的实例演示了这点：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 响应式的导航栏</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<nav class="navbar navbar-default" role="navigation">
   <div class="navbar-header">
      <button type="button" class="navbar-toggle" data-toggle="col
         data-target="#example-navbar-collapse">
         <span class="sr-only">切换导航</span>
         <span class="icon-bar"></span>
         <span class="icon-bar"></span>
         <span class="icon-bar"></span>
      </button>
      <a class="navbar-brand" href="#">W3Cschool</a>
   </div>
   <div class="collapse navbar-collapse" id="example-navbar-collaps
      <ul class="nav navbar-nav">
         <li class="active"><a href="#">iOS</a></li>
         <li><a href="#">SVN</a></li>
         <li class="dropdown">
            <a href="#" class="dropdown-toggle" data-toggle="dropdo
               Java <b class="caret"></b>
            </a>
            <ul class="dropdown-menu">
               <li><a href="#">jmeter</a></li>
               <li><a href="#">EJB</a></li>
               <li><a href="#">Jasper Report</a></li>
               <li class="divider"></li>
               <li><a href="#">分离的链接</a></li>
               <li class="divider"></li>
               <li><a href="#">另一个分离的链接</a></li>
            </ul>
         </li>
      </ul>
   </div>
</nav>

</body>
</html>
```

结果如下所示：



导航栏中的表单

导航栏中的表单不是使用 Bootstrap 表单 章节中所讲到的默认的 class，它是使用 **.navbar-form** class。这确保了表单适当的垂直对齐和在较窄的视口中折叠的行为。使用对齐方式选项（这将在组件对齐方式部分进行详细讲解）来决定导航栏中的内容放置在哪里。

下面的实例演示了这点：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 导航栏中的表单</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<nav class="navbar navbar-default" role="navigation">
   <div class="navbar-header">
      <a class="navbar-brand" href="#">W3Cschool</a>
   </div>
   <div>
      <form class="navbar-form navbar-left" role="search">
         <div class="form-group">
            <input type="text" class="form-control" placeholder="Se
         </div>
         <button type="submit" class="btn btn-default">提交</button
      </form>
   </div>
</nav>

</body>
</html>
```

结果如下所示：

W3Cschool Search 提交

导航栏中的按钮

您可以使用 class **.navbar-btn** 向不在 <form> 中的 <button> 元素添加按钮，按钮在导航栏上垂直居中。**.navbar-btn** 可被使用在 <a> 和 <input> 元素上。

> 不要在 **.navbar-nav** 内的 <a> 元素上使用 **.navbar-btn**，因为它不是标准的 button class。

下面的实例演示了这点：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 导航栏中的按钮</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<nav class="navbar navbar-default" role="navigation">
   <div class="navbar-header">
      <a class="navbar-brand" href="#">W3Cschool</a>
   </div>
   <div>
      <form class="navbar-form navbar-left" role="search">
         <div class="form-group">
            <input type="text" class="form-control" placeholder="Se
         </div>
         <button type="submit" class="btn btn-default">提交按钮</but
      </form>
      <button type="button" class="btn btn-default navbar-btn">
         导航栏按钮
      </button>
   </div>
</nav>

</body>
</html>
```

结果如下所示：

W3Cschool Search 提交按钮 导航栏按钮

导航栏中的文本

如果需要在导航中包含文本字符串，请使用 class **.navbar-text**。这通常与 <p> 标签一起使用，确保适当的前导和颜色。下面的实例演示了这点：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 导航栏中的文本</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<nav class="navbar navbar-default" role="navigation">
   <div class="navbar-header">
      <a class="navbar-brand" href="#">W3Cschool</a>
   </div>
   <div>
      <p class="navbar-text">Signed in as Thomas</p>
   </div>
</nav>

</body>
</html>
```

结果如下所示：

W3Cschool　Signed in as Thomas

非导航链接

如果您不想在常规的导航栏导航组件内使用标准的链接，那么请使用 class **navbar-link** 来为默认的和倒转的导航栏选项添加适当的颜色，如下面的实例所示：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 非导航链接</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<nav class="navbar navbar-default" role="navigation">
   <div class="navbar-header">
      <a class="navbar-brand" href="#">W3Cschool</a>
   </div>
   <div>
      <p class="navbar-text navbar-right">Signed in as
         <a href="#" class="navbar-link">Thomas</a>
      </p>
   </div>
</div>

</body>
</html>
```

结果如下所示：

W3Cschool Signed in as Thomas

组件对齐方式

您可以使用实用工具 class **.navbar-left** 或 **.navbar-right** 向左或向右对齐导航栏中的 导航链接、表单、按钮或文本 这些组件。这两个 class 都会在指定的方向上添加 CSS 浮动。下面的实例演示了这点：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 组件对齐方式</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<nav class="navbar navbar-default" role="navigation">
   <div class="navbar-header">
      <a class="navbar-brand" href="#">W3Cschool</a>
```

```
    </div>
    <div>
       <!--向左对齐-->
       <ul class="nav navbar-nav navbar-left">
          <li class="dropdown">
             <a href="#" class="dropdown-toggle" data-toggle="dropdo
                Java
                <b class="caret"></b>
             </a>
             <ul class="dropdown-menu">
                <li><a href="#">jmeter</a></li>
                <li><a href="#">EJB</a></li>
                <li><a href="#">Jasper Report</a></li>
                <li class="divider"></li>
                <li><a href="#">分离的链接</a></li>
                <li class="divider"></li>
                <li><a href="#">另一个分离的链接</a></li>
             </ul>
          </li>
       </ul>
       <form class="navbar-form navbar-left" role="search">
          <button type="submit" class="btn btn-default">
             向左对齐-提交按钮
          </button>
       </form>
       <p class="navbar-text navbar-left">向左对齐-文本</p>
       <!--向右对齐-->
       <ul class="nav navbar-nav navbar-right">
          <li class="dropdown">
             <a href="#" class="dropdown-toggle" data-toggle="dropdo
                Java <b class="caret"></b>
             </a>
             <ul class="dropdown-menu">
                <li><a href="#">jmeter</a></li>
                <li><a href="#">EJB</a></li>
                <li><a href="#">Jasper Report</a></li>
                <li class="divider"></li>
                <li><a href="#">分离的链接</a></li>
                <li class="divider"></li>
                <li><a href="#">另一个分离的链接</a></li>
             </ul>
          </li>
       </ul>
       <form class="navbar-form navbar-right" role="search">
          <button type="submit" class="btn btn-default">
             向右对齐-提交按钮
          </button>
       </form>
       <p class="navbar-text navbar-right">向右对齐-文本</p>
    </div>
 </nav>

 </body>
```

```
</html>
```

结果如下所示：



固定到顶部

Bootstrap 导航栏可以动态定位。默认情况下，它是块级元素，它是基于在 HTML 中放置的位置定位的。通过一些帮助器类，您可以把它放置在页面的顶部或者底部，或者您可以让它成为随着页面一起滚动的静态导航栏。

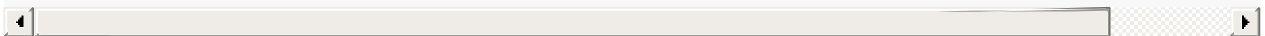
如果您想要让导航栏固定在页面的顶部，请向 **.navbar class** 添加 class **.navbar-fixed-top**。下面的实例演示了这点：

> 为了防止导航栏与页面主体中的其他内容的顶部相交错，请向 <body> 标签添加至少 50 像素的内边距（padding），内边距的值可以根据您的需要进行设置。

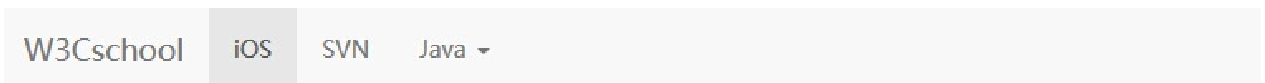
```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 固定到顶部</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<nav class="navbar navbar-default navbar-fixed-top" role="navigati
   <div class="navbar-header">
      <a class="navbar-brand" href="#">W3Cschool</a>
   </div>
   <div>
      <ul class="nav navbar-nav">
         <li class="active"><a href="#">iOS</a></li>
         <li><a href="#">SVN</a></li>
         <li class="dropdown">
            <a href="#" class="dropdown-toggle" data-toggle="dropd
               Java <b class="caret"></b>
            </a>
            <ul class="dropdown-menu">
               <li><a href="#">jmeter</a></li>
               <li><a href="#">EJB</a></li>
               <li><a href="#">Jasper Report</a></li>
               <li class="divider"></li>
               <li><a href="#">分离的链接</a></li>
               <li class="divider"></li>
               <li><a href="#">另一个分离的链接</a></li>
            </ul>
         </li>
      </ul>
   </div>
</nav>

</body>
</html>
```



结果如下所示：



固定到底部

如果您想要让导航栏固定在页面的底部，请向 **.navbar class** 添加 class **.navbar-fixed-bottom**。下面的实例演示了这点：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 固定到底部</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<nav class="navbar navbar-default navbar-fixed-bottom" role="naviga
   <div class="navbar-header">
      <a class="navbar-brand" href="#">W3Cschool</a>
   </div>
   <div>
      <ul class="nav navbar-nav">
         <li class="active"><a href="#">iOS</a></li>
         <li><a href="#">SVN</a></li>
         <li class="dropdown">
            <a href="#" class="dropdown-toggle" data-toggle="dropdo
               Java <b class="caret"></b>
            </a>
            <ul class="dropdown-menu">
               <li><a href="#">jmeter</a></li>
               <li><a href="#">EJB</a></li>
               <li><a href="#">Jasper Report</a></li>
               <li class="divider"></li>
               <li><a href="#">分离的链接</a></li>
               <li class="divider"></li>
               <li><a href="#">另一个分离的链接</a></li>
            </ul>
         </li>
      </ul>
   </div>
</nav>

</body>
</html>
```

结果如下所示：

W3Cschool iOS SVN Java

静态的顶部

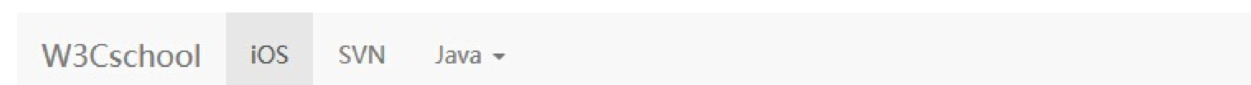
如需创建能随着页面一起滚动的导航栏，请添加 **.navbar-static-top** class。该 class 不要求向 <body> 添加内边距（padding）。

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 静态的顶部</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<nav class="navbar navbar-default navbar-static-top" role="navigati
   <div class="navbar-header">
      <a class="navbar-brand" href="#">W3Cschool</a>
   </div>
   <div>
      <ul class="nav navbar-nav">
         <li class="active"><a href="#">iOS</a></li>
         <li><a href="#">SVN</a></li>
         <li class="dropdown">
            <a href="#" class="dropdown-toggle" data-toggle="dropdo
               Java <b class="caret"></b>
            </a>
            <ul class="dropdown-menu">
               <li><a href="#">jmeter</a></li>
               <li><a href="#">EJB</a></li>
               <li><a href="#">Jasper Report</a></li>
               <li class="divider"></li>
               <li><a href="#">分离的链接</a></li>
               <li class="divider"></li>
               <li><a href="#">另一个分离的链接</a></li>
            </ul>
         </li>
      </ul>
   </div>
</nav>

</body>
</html>
```

结果如下所示：



倒置的导航栏

为了创建一个带有黑色背景白色文本的倒置的导航栏，只需要简单地向 **.navbar** class 添加 **.navbar-inverse** class 即可，如下面的实例所示：

> 为了防止导航栏与页面主体中的其他内容的顶部相交错，请向 <body> 标签添加至少 50 像素的内边距（padding），内边距的值可以根据您的需要进行设置。

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 倒置的导航栏</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<nav class="navbar navbar-inverse" role="navigation">
   <div class="navbar-header">
      <a class="navbar-brand" href="#">W3Cschool</a>
   </div>
   <div>
      <ul class="nav navbar-nav">
         <li class="active"><a href="#">iOS</a></li>
         <li><a href="#">SVN</a></li>
         <li class="dropdown">
            <a href="#" class="dropdown-toggle" data-toggle="dropdo
               Java <b class="caret"></b>
            </a>
            <ul class="dropdown-menu">
               <li><a href="#">jmeter</a></li>
               <li><a href="#">EJB</a></li>
               <li><a href="#">Jasper Report</a></li>
               <li class="divider"></li>
               <li><a href="#">分离的链接</a></li>
               <li class="divider"></li>
               <li><a href="#">另一个分离的链接</a></li>
            </ul>
         </li>
      </ul>
   </div>
</nav>

</body>
</html>
```

结果如下所示：



Bootstrap 面包屑导航（Breadcrumbs）

面包屑导航（Breadcrumbs）是一种基于网站层次信息的显示方式。以博客为例，面包屑导航可以显示发布日期、类别或标签。它们表示当前页面在导航层次结构内的位置。

Bootstrap 中的面包屑导航（Breadcrumbs）是一个简单的带有 **.breadcrumb** class 的无序列表。分隔符会通过 CSS（bootstrap.min.css）中下面所示的 class 自动被添加：

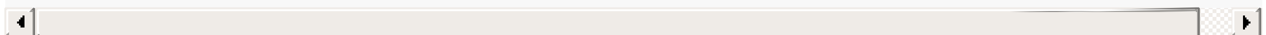
```
.breadcrumb > li + li:before {
    color: #CCCCCC;
    content: "/ ";
    padding: 0 5px;
}
```

下面的实例演示了面包屑导航：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 面包屑导航</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<ol class="breadcrumb">
  <li><a href="#">Home</a></li>
  <li><a href="#">2013</a></li>
  <li class="active">十一月</li>
</ol>

</body>
</html>
```



结果如下所示：

Home / 2013 / 十一月

Bootstrap 分页

本章将讲解 Bootstrap 支持的分页特性。分页（Pagination），是一种无序列表，Bootstrap 像处理其他界面元素一样处理分页。

分页（Pagination）

下表列出了 Bootstrap 提供的处理分页的 class。

.pagination

添加该 class 来在页面上显示分页。

```
&lt;ul class="pagination"&gt;
  &lt;li&gt;&lt;a href="#"&gt;&laquo;&lt;/a&gt;&lt;/li&gt;
  &lt;li&gt;&lt;a href="#"&gt;1&lt;/a&gt;&lt;/li&gt;
  .......
&lt;/ul&gt;
```

.disabled, .active

您可以自定义链接，通过使用 **.disabled** 来定义不可点击的链接，通过使用 **.active** 来指示当前的页面。

```
&lt;ul class="pagination"&gt;
  &lt;li class="disabled"&gt;&lt;a href="#"&gt;&laquo;&lt;/a&gt;&lt
  &lt;li class="active"&gt;&lt;a href="#"&gt;1&lt;span class="sr-on
  .......
&lt;/ul&gt;
```



.pagination-lg, .pagination-sm

使用这些 class 来获取不同大小的项。

```
&lt;ul class="pagination pagination-lg"&gt;...&lt;/ul&gt;
&lt;ul class="pagination"&gt;...&lt;/ul&gt;
&lt;ul class="pagination pagination-sm"&gt;...&lt;/ul&gt;
```

默认的分页

下面的实例演示了上表中所讨论的 class **.pagination** 的用法：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 默认的分页</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<ul class="pagination">
  <li><a href="#">&laquo;</a></li>
  <li><a href="#">1</a></li>
  <li><a href="#">2</a></li>
  <li><a href="#">3</a></li>
  <li><a href="#">4</a></li>
  <li><a href="#">5</a></li>
  <li><a href="#">&raquo;</a></li>
</ul>

</body>
</html>
```

结果如下所示：



分页的状态

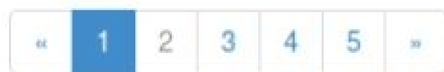
下面的实例演示了上表中所讨论的 class **.disabled**、**.active** 的用法：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 分页的状态</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<ul class="pagination">
  <li><a href="#">&laquo;</a></li>
  <li class="active"><a href="#">1</a></li>
  <li class="disabled"><a href="#">2</a></li>
  <li><a href="#">3</a></li>
  <li><a href="#">4</a></li>
  <li><a href="#">5</a></li>
  <li><a href="#">&raquo;</a></li>
</ul>

</body>
</html>
```

结果如下所示：



分页的大小

下面的实例演示了上表中所讨论的 class **.pagination-*** 的用法：

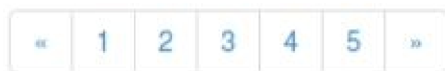
```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 分页的大小</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<ul class="pagination pagination-lg">
  <li><a href="#">&laquo;</a></li>
  <li><a href="#">1</a></li>
  <li><a href="#">2</a></li>
  <li><a href="#">3</a></li>
  <li><a href="#">4</a></li>
  <li><a href="#">5</a></li>
  <li><a href="#">&raquo;</a></li>
</ul><br>
<ul class="pagination">
  <li><a href="#">&laquo;</a></li>
  <li><a href="#">1</a></li>
  <li><a href="#">2</a></li>
  <li><a href="#">3</a></li>
  <li><a href="#">4</a></li>
  <li><a href="#">5</a></li>
  <li><a href="#">&raquo;</a></li>
</ul><br>
<ul class="pagination pagination-sm">
 <li><a href="#">&laquo;</a></li>
  <li><a href="#">1</a></li>
  <li><a href="#">2</a></li>
  <li><a href="#">3</a></li>
  <li><a href="#">4</a></li>
  <li><a href="#">5</a></li>
  <li><a href="#">&raquo;</a></li>
</ul>

</body>
</html>
```

结果如下所示：







翻页（Pager）

如果您想要创建一个简单的分页链接为用户提供导航，可通过翻页来实现。与分页链接一样，翻页也是无序列表。默认情况下，链接是居中显示。下表列出了 Bootstrap 处理翻页的 class。

.pager

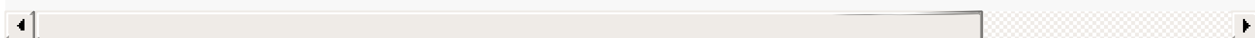
添加该 class 来获得翻页链接。

```
&lt;ul class="pager"&gt;
  &lt;li&gt;&lt;a href="#"&gt;Previous&lt;/a&gt;&lt;/li&gt;
  &lt;li&gt;&lt;a href="#"&gt;Next&lt;/a&gt;&lt;/li&gt;
&lt;/ul&gt;
```

.previous, .next

使用 class **.previous** 把链接向左对齐，使用 **.next** 把链接向右对齐。

```
&lt;ul class="pager"&gt;
  &lt;li class="previous"&gt;&lt;a href="#"&gt;&larr; Older&lt;/a&
  &lt;li class="next"&gt;&lt;a href="#"&gt;Newer &rarr;&lt;/a&gt;&l
&lt;/ul&gt;
```



.disabled

添加该 class 来获得一个颜色变淡的外观。

```
&lt;ul class="pager"&gt;
  &lt;li class="previous disabled"&gt;&lt;a href="#"&gt;&larr; Olde
  &lt;li class="next"&gt;&lt;a href="#"&gt;Newer &rarr;&lt;/a&gt;&l
&lt;/ul&gt;
```

默认的翻页

下面的实例演示了上表中所讨论的 class **.pager** 的用法：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 默认的翻页</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<ul class="pager">
  <li><a href="#">Previous</a></li>
  <li><a href="#">Next</a></li>
</ul>

</body>
</html>
```

结果如下所示：



对齐的链接

下面的实例演示了上表中所讨论的 class **.previous**、**.next** 的用法：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 翻页中对齐的链接</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<ul class="pager">
  <li class="previous"><a href="#">&larr; Older</a></li>
  <li class="next"><a href="#">Newer &rarr;</a></li>
</ul>

</body>
</html>
```

结果如下所示：



翻页的状态

下面的实例演示了上表中所讨论的 class **.disabled** 的用法：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 翻页的状态</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<ul class="pager">
  <li class="previous disabled"><a href="#">&larr; Older</a></li>
  <li class="next"><a href="#">Newer &rarr;</a></li>
</ul>

</body>
</html>
```

结果如下所示：



Bootstrap 标签

本章将讲解 Bootstrap 标签。标签可用于计数、提示或页面上其他的标记显示。使用 class **.label** 来显示标签，如下面的实例所示：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 标签</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<h1>Example Heading <span class="label label-default">Label</span><
<h2>Example Heading <span class="label label-default">Label</span><
<h3>Example Heading <span class="label label-default">Label</span><
<h4>Example Heading <span class="label label-default">Label</span><

</body>
</html>
```

结果如下所示：



您可以使用修饰的 class **label-default**、**label-primary**、**label-success**、**label-info**、**label-warning**、**label-danger** 来改变标签的外观，如下面的实例所示：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 标签的变体</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<span class="label label-default">默认标签</span>
<span class="label label-primary">主要标签</span>
<span class="label label-success">成功标签</span>
<span class="label label-info">信息标签</span>
<span class="label label-warning">警告标签</span>
<span class="label label-danger">危险标签</span>

</body>
</html>
```

结果如下所示：

默认标签 主要标签 成功标签 信息标签 警告标签 危险标签

Bootstrap 徽章（Badges）

本章将讲解 Bootstrap 徽章（Badges）。徽章与标签相似，主要的区别在于徽章的边角更加圆滑。

徽章（Badges）主要用于突出显示新的或未读的项。如需使用徽章，只需要把 **<span class="badge">** 添加到链接、Bootstrap 导航等这些元素上即可。

下面的实例演示了这点：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 徽章（Badges）</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<a href="#">Mailbox <span class="badge">50</span></a>

</body>
</html>
```

结果如下所示：

Mailbox 50

当没有新的或未读的项时，通过 CSS 的 **:empty** 选择器，徽章会折叠起来，表示里边没有内容。

激活导航状态

您可以在激活状态的胶囊式导航和列表导航中放置徽章。通过使用 **<span class="badge">** 来激活链接，如下面的实例所示：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 激活导航状态</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<h4>胶囊式导航中的激活状态</h4>
<ul class="nav nav-pills">
   <li class="active"><a href="#">首页 <span class="badge">42</span
   <li><a href="#">简介</a></li>
   <li><a href="#">消息 <span class="badge">3</span></a></li>
</ul>
<br>
<h4>列表导航中的激活状态</h4>
<ul class="nav nav-pills nav-stacked" style="max-width: 260px;">
   <li class="active">
      <a href="#">
         <span class="badge pull-right">42</span>
         首页
      </a>
   </li>
   <li><a href="#">简介</a></li>
   <li>
      <a href="#">
         <span class="badge pull-right">3</span>
         消息
      </a>
   </li>
</ul>

</body>
</html>
```

结果如下所示：

胶囊式导航中的激活状态



列表导航中的激活状态



Bootstrap 超大屏幕（Jumbotron）

本章将讲解 Bootstrap 支持的另一个特性，超大屏幕（Jumbotron）。顾名思义该组件可以增加标题的大小，并为登陆页面内容添加更多的外边距（margin）。使用超大屏幕（Jumbotron）的步骤如下：

- 创建一个带有 class **.jumbotron**. 的容器 <div>。
- 除了更大的 <h1>，字体粗细 *font-weight* 被减为 200px。

下面的实例演示了这点：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 超大屏幕（Jumbotron）</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="container">
   <div class="jumbotron">
      <h1>欢迎登陆页面！</h1>
      <p>这是一个超大屏幕（Jumbotron）的实例。</p>
      <p><a class="btn btn-primary btn-lg" role="button">
         学习更多</a>
      </p>
   </div>
</div>

</body>
</html>
```

结果如下所示：

欢迎登陆页面！

这是一个超大屏幕（Jumbotron）的实例。

学习更多

为了获得占用全部宽度且不带圆角的超大屏幕，请在所有的 **.container** class 外使用 **.jumbotron** class，如下面的实例所示：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 超大屏幕（Jumbotron）</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="jumbotron">
   <div class="container">
      <h1>欢迎登陆页面！</h1>
      <p>这是一个超大屏幕（Jumbotron）的实例。</p>
      <p><a class="btn btn-primary btn-lg" role="button">
         学习更多</a>
      </p>
   </div>
</div>

</body>
</html>
```

结果如下所示：

欢迎登陆页面！

这是一个超大屏幕（Jumbotron）的实例。

学习更多

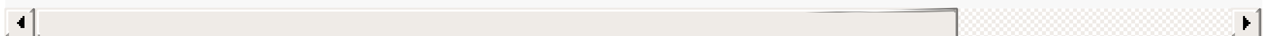
Bootstrap 页面标题（Page Header）

页面标题（Page Header）是个不错的功能，它会在网页标题四周添加适当的间距。当一个网页中有多个标题且每个标题之间需要添加一定的间距时，页面标题这个功能就显得特别有用。如需使用页面标题（Page Header），请把您的标题放置在带有 class **.page-header** 的 <div> 中：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 页面标题</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="page-header">
   <h1>页面标题实例
      <small>子标题</small>
   </h1>
</div>
<p>这是一个示例文本。这是一个示例文本。这是一个示例文本。这是一个示例文本。这是

</body>
</html>
```



结果如下所示：

页面标题实例 子标题

这是一个示例文本。这是一个示例文本。这是一个示例文本。这是一个示例文本。这是一个示例文本。

Bootstrap 缩略图

本章将讲解 Bootstrap 缩略图。大多数站点都需要在网格中布局图像、视频、文本等。Bootstrap 通过缩略图为此提供了一种简便的方式。使用 Bootstrap 创建缩略图的步骤如下：

- 在图像周围添加带有 class **.thumbnail** 的 <a> 标签。
- 这会添加四个像素的内边距（padding）和一个灰色的边框。
- 当鼠标悬停在图像上时，会动画显示出图像的轮廓。

下面的实例演示了默认的缩略图：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 缩略图</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="row">
   <div class="col-sm-6 col-md-3">
      <a href="#" class="thumbnail">
         <img src="img/kittens.jpg"
         alt="通用的占位符缩略图">
      </a>
   </div>
   <div class="col-sm-6 col-md-3">
      <a href="#" class="thumbnail">
         <img src="img/kittens.jpg"
         alt="通用的占位符缩略图">
      </a>
   </div>
   <div class="col-sm-6 col-md-3">
      <a href="#" class="thumbnail">
         <img src="img/kittens.jpg"
         alt="通用的占位符缩略图">
      </a>
   </div>
   <div class="col-sm-6 col-md-3">
      <a href="#" class="thumbnail">
         <img src="img/kittens.jpg"
         alt="通用的占位符缩略图">
      </a>
   </div>
</div>

</body>
</html>
```

结果如下所示：

添加自定义的内容

现在我们有了一个基本的缩略图，我们可以向缩略图添加各种 HTML 内容，比如标题、段落或按钮。具体步骤如下：

- 把带有 class **.thumbnail** 的 <a> 标签改为 <div>。
- 在该 <div> 内，您可以添加任何您想要添加的东西。由于这是一个 <div>，我们可以使用默认的基于 span 的命名规则来调整大小。
- 如果您想要给多个图像进行分组，请把它们放置在一个无序列表中，且每个列表项向左浮动。

下面的实例演示了这点：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 自定义缩略图</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="row">
   <div class="col-sm-6 col-md-3">
      <div class="thumbnail">
         <img src="img/kittens.jpg"
         alt="通用的占位符缩略图">
      </div>
      <div class="caption">
         <h3>缩略图标签</h3>
         <p>一些示例文本。一些示例文本。</p>
         <p>
            <a href="#" class="btn btn-primary" role="button">
               按钮
            </a>
            <a href="#" class="btn btn-default" role="button">
               按钮
            </a>
         </p>
      </div>
   </div>
   <div class="col-sm-6 col-md-3">
      <div class="thumbnail">
         <img src="img/kittens.jpg"
         alt="通用的占位符缩略图">
      </div>
      <div class="caption">
         <h3>缩略图标签</h3>
         <p>一些示例文本。一些示例文本。</p>
         <p>
```

```
            <a href="#" class="btn btn-primary" role="button">
               按钮
            </a>
            <a href="#" class="btn btn-default" role="button">
               按钮
            </a>
         </p>
      </div>
   </div>
   <div class="col-sm-6 col-md-3">
      <div class="thumbnail">
         <img src="img/kittens.jpg"
         alt="通用的占位符缩略图">
      </div>
      <div class="caption">
         <h3>缩略图标签</h3>
         <p>一些示例文本。一些示例文本。</p>
         <p>
            <a href="#" class="btn btn-primary" role="button">
               按钮
            </a>
            <a href="#" class="btn btn-default" role="button">
               按钮
            </a>
         </p>
      </div>
   </div>
   <div class="col-sm-6 col-md-3">
      <div class="thumbnail">
         <img src="img/kittens.jpg"
         alt="通用的占位符缩略图">
      </div>
      <div class="caption">
         <h3>缩略图标签</h3>
         <p>一些示例文本。一些示例文本。</p>
         <p>
            <a href="#" class="btn btn-primary" role="button">
               按钮
            </a>
            <a href="#" class="btn btn-default" role="button">
               按钮
            </a>
         </p>
      </div>
   </div>
</div>

</body>
</html>
```

结果如下所示：

缩略图标签

一些示例文本。一些示例文本。

按钮 按钮

缩略图标签

一些示例文本。一些示例文本。

按钮 按钮

缩略图标签

一些示例文本。一些示例文本。

按钮 按钮

缩略图标签

一些示例文本。一些示例文本。

按钮 按钮

Bootstrap 警告（Alerts）

本章将讲解警告（Alerts）以及 Bootstrap 所提供的用于警告的 class。警告（Alerts）向用户提供了一种定义消息样式的方式。它们为典型的用户操作提供了上下文信息反馈。

您可以为警告框添加一个可选的关闭按钮。为了创建一个内联的可取消的警告框，请使用 警告（Alerts） jQuery 插件。

您可以通过创建一个 <div>，并向其添加一个 **.alert** class 和四个上下文 class（即 **.alert-success**、**.alert-info**、**.alert-warning**、**.alert-danger**）之一，来添加一个基本的警告框。下面的实例演示了这点：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 警告（Alerts）</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="alert alert-success">成功！很好地完成了提交。</div>
<div class="alert alert-info">信息！请注意这个信息。</div>
<div class="alert alert-warning">警告！请不要提交。</div>
<div class="alert alert-danger">错误！请进行一些更改。</div>

</body>
</html>
```

结果如下所示：

成功！很好地完成了提交。

信息！请注意这个信息。

警告！请不要提交。

错误！请进行一些更改。

可取消的警告（Dismissal Alerts）

创建一个可取消的警告（Dismissal Alert）步骤如下：

- 通过创建一个 <div>，并向其添加一个 **.alert** class 和四个上下文 class（即 **.alert-success**、**.alert-info**、**.alert-warning**、**.alert-danger**）之一，来添加一个基本的警告框。
- 同时向上面的 <div> class 添加可选的 **.alert-dismissable**。
- 添加一个关闭按钮。

下面的实例演示了这点：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 可取消的警告（Dismissal Alerts）</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="alert alert-success alert-dismissable">
   <button type="button" class="close" data-dismiss="alert"
      aria-hidden="true">
      &times;
   </button>
   成功！很好地完成了提交。
</div>
<div class="alert alert-info alert-dismissable">
   <button type="button" class="close" data-dismiss="alert"
      aria-hidden="true">
      &times;
   </button>
   信息！请注意这个信息。
</div>
<div class="alert alert-warning alert-dismissable">
   <button type="button" class="close" data-dismiss="alert"
      aria-hidden="true">
      &times;
   </button>
   警告！请不要提交。
</div>
<div class="alert alert-danger alert-dismissable">
   <button type="button" class="close" data-dismiss="alert"
      aria-hidden="true">
      &times;
   </button>
   错误！请进行一些更改。
</div>

</body>
</html>
```

> 请确保使用带有 *data-dismiss="alert"* data 属性的 <button> 元素。

结果如下所示：

成功！很好地完成了提交。 

信息！请注意这个信息。 

警告！请不要提交。 

错误！请进行一些更改。 

警告（Alerts）中的链接

在警告（Alerts）中创建链接的步骤如下：

- 通过创建一个 <div>，并向其添加一个 **.alert** class 和四个上下文 class（即 **.alert-success**、**.alert-info**、**.alert-warning**、**.alert-danger**）之一，来添加一个基本的警告框。
- 使用 **.alert-link** 实体类来快速提供带有匹配颜色的链接。

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 警告（Alerts）中的链接</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="alert alert-success">
   <a href="#" class="alert-link">成功！很好地完成了提交。</a>
</div>
<div class="alert alert-info">
   <a href="#" class="alert-link">信息！请注意这个信息。</a>
</div>
<div class="alert alert-warning">
   <a href="#" class="alert-link">警告！请不要提交。</a>
</div>
<div class="alert alert-danger">
   <a href="#" class="alert-link">错误！请进行一些更改。</a>
</div>

</body>
</html>
```

结果如下所示：

成功！很好地完成了提交。

信息！请注意这个信息。

警告！请不要提交。

错误！请进行一些更改。

Bootstrap 进度条

本章将讲解 Bootstrap 进度条。在本教程中，您将看到如何使用 Bootstrap 创建加载、重定向或动作状态的进度条。

> Bootstrap 进度条使用 CSS3 过渡和动画来获得该效果。Internet Explorer 9 及之前的版本和旧版的 Firefox 不支持该特性，Opera 12 不支持动画。

默认的进度条

创建一个基本的进度条的步骤如下：

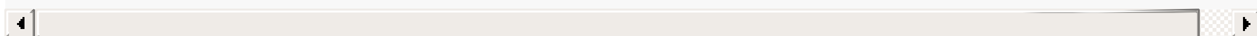
- 添加一个带有 class **.progress** 的 <div>。
- 接着，在上面的 <div> 内，添加一个带有 class **.progress-bar** 的空的 <div>。
- 添加一个带有百分比表示的宽度的 style 属性，例如 style="60%"; 表示进度条在 60% 的位置。

让我们看看下面的实例：

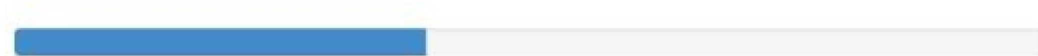
```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 进度条</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="progress">
   <div class="progress-bar" role="progressbar" aria-valuenow="60"
      aria-valuemin="0" aria-valuemax="100" style="width: 40%;">
      <span class="sr-only">40% 完成</span>
   </div>
</div>

</body>
</html>
```



结果如下所示：



交替的进度条

创建不同样式的进度条的步骤如下：

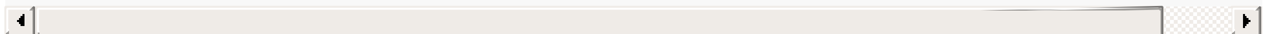
- 添加一个带有 class **.progress** 的 <div>。
- 接着，在上面的 <div> 内，添加一个带有 class **.progress-bar** 和 class **progress-bar-*** 的空的 <div>。其中， 可以是 **success*、*info*、*warning*、*danger*。
- 添加一个带有百分比表示的宽度的 style 属性，例如 style="60%"; 表示进度条在 60% 的位置。

让我们看看下面的实例：

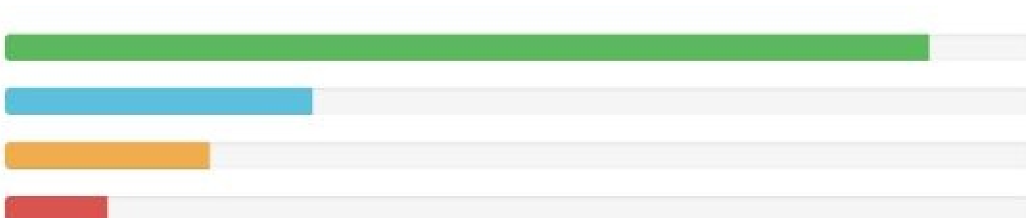
```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 交替的进度条</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="progress">
   <div class="progress-bar progress-bar-success" role="progressbar
      aria-valuenow="60" aria-valuemin="0" aria-valuemax="100"
      style="width: 90%;">
      <span class="sr-only">90% 完成（成功）</span>
   </div>
</div>
<div class="progress">
   <div class="progress-bar progress-bar-info" role="progressbar"
      aria-valuenow="60" aria-valuemin="0" aria-valuemax="100"
      style="width: 30%;">
      <span class="sr-only">30% 完成（信息）</span>
   </div>
</div>
<div class="progress">
   <div class="progress-bar progress-bar-warning" role="progressbar
      aria-valuenow="60" aria-valuemin="0" aria-valuemax="100"
      style="width: 20%;">
      <span class="sr-only">20% 完成（警告）</span>
   </div>
</div>
<div class="progress">
   <div class="progress-bar progress-bar-danger" role="progressbar
      aria-valuenow="60" aria-valuemin="0" aria-valuemax="100"
      style="width: 10%;">
      <span class="sr-only">10% 完成（危险）</span>
   </div>
</div>

</body>
</html>
```



结果如下所示：



条纹的进度条

创建一个条纹的进度条的步骤如下：

- 添加一个带有 class **.progress** 和 **.progress-striped** 的 <div>。
- 接着，在上面的 <div> 内，添加一个带有 class **.progress-bar** 和 class **progress-bar-*** 的空的 <div>。其中， 可以是 **success*、*info*、*warning*、*danger*。
- 添加一个带有百分比表示的宽度的 style 属性，例如 style="60%"; 表示进度条在 60% 的位置。

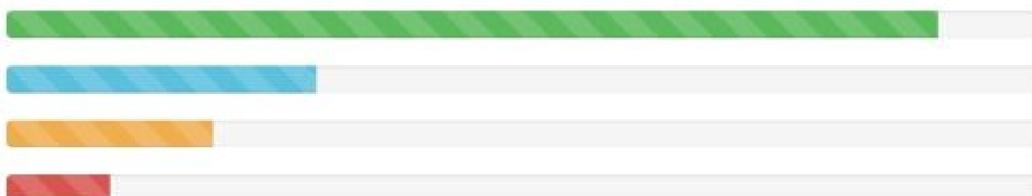
让我们看看下面的实例：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 条纹的进度条</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="progress progress-striped">
   <div class="progress-bar progress-bar-success" role="progressbar
      aria-valuenow="60" aria-valuemin="0" aria-valuemax="100"
      style="width: 90%;">
      <span class="sr-only">90% 完成（成功）</span>
   </div>
</div>
<div class="progress progress-striped">
   <div class="progress-bar progress-bar-info" role="progressbar"
      aria-valuenow="60" aria-valuemin="0" aria-valuemax="100"
      style="width: 30%;">
      <span class="sr-only">30% 完成（信息）</span>
   </div>
</div>
<div class="progress progress-striped">
   <div class="progress-bar progress-bar-warning" role="progressbar
      aria-valuenow="60" aria-valuemin="0" aria-valuemax="100"
      style="width: 20%;">
      <span class="sr-only">20% 完成（警告）</span>
   </div>
</div>
<div class="progress progress-striped">
   <div class="progress-bar progress-bar-danger" role="progressbar'
      aria-valuenow="60" aria-valuemin="0" aria-valuemax="100"
      style="width: 10%;">
      <span class="sr-only">10% 完成（危险）</span>
   </div>
</div>

</body>
</html>
```

结果如下所示：



动画的进度条

创建一个动画的进度条的步骤如下：

- 添加一个带有 class **.progress** 和 **.progress-striped** 的 <div>。同时添加 class **.active**。
- 接着，在上面的 <div> 内，添加一个带有 class **.progress-bar** 的空的 <div>。
- 添加一个带有百分比表示的宽度的 style 属性，例如 style="60%"; 表示进度条在 60% 的位置。

这将会使条纹具有从右向左的运动感。

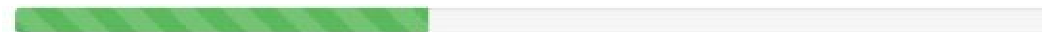
让我们看看下面的实例：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 动画的进度条</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="progress progress-striped active">
   <div class="progress-bar progress-bar-success" role="progressbar
      aria-valuenow="60" aria-valuemin="0" aria-valuemax="100"
      style="width: 40%;">
      <span class="sr-only">40% 完成</span>
   </div>
</div>

</body>
</html>
```

结果如下所示：



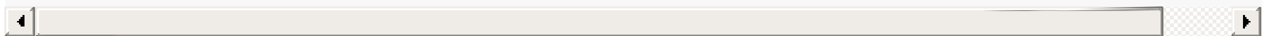
堆叠的进度条

您甚至可以堆叠多个进度条。把多个进度条放在相同的 **.progress** 中即可实现堆叠，如下面的实例所示：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 堆叠的进度条</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="progress">
   <div class="progress-bar progress-bar-success" role="progressbar
      aria-valuenow="60" aria-valuemin="0" aria-valuemax="100"
      style="width: 40%;">
      <span class="sr-only">40% 完成</span>
   </div>
   <div class="progress-bar progress-bar-info" role="progressbar"
      aria-valuenow="60" aria-valuemin="0" aria-valuemax="100"
      style="width: 30%;">
      <span class="sr-only">30% 完成（信息）</span>
   </div>
   <div class="progress-bar progress-bar-warning" role="progressbar
      aria-valuenow="60" aria-valuemin="0" aria-valuemax="100"
      style="width: 20%;">
      <span class="sr-only">20% 完成（警告）</span>
   </div>
</div>

</body>
</html>
```



结果如下所示：



Bootstrap 多媒体对象（Media Object）

本章我们将讲解 Bootstrap 中的多媒体对象（Media Object）。这些抽象的对象样式用于创建各种类型的组件（比如：博客评论），我们可以在组件中使用图文混排，图像可以左对齐或者右对齐。媒体对象可以用更少的代码来实现媒体对象与文字的混排。

媒体对象轻量标记、易于扩展的特性是通过向简单的标记应用 class 来实现的。你可以在 HTML 标签中添加以下两种形式来设置媒体对象：

- **.media**：该 class 允许将媒体对象里的多媒体（图像、视频、音频）浮动到内容区块的左边或者右边。
- **.media-list**：如果你需要一个列表，各项内容是无序列表的一部分，可以使用该 class。可用于评论列表与文章列表。

让我们来看看下面这个有关默认的媒体对象 .media 的实例：

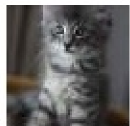
```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 默认的媒体对象</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="media">
   <a class="pull-left" href="#">
      <img class="media-object" src="img/64.jpg"
      alt="媒体对象">
   </a>
   <div class="media-body">
      <h4 class="media-heading">媒体标题</h4>
      这是一些示例文本。这是一些示例文本。
      这是一些示例文本。这是一些示例文本。
      这是一些示例文本。这是一些示例文本。
      这是一些示例文本。这是一些示例文本。
      这是一些示例文本。这是一些示例文本。
   </div>
</div>
<div class="media">
   <a class="pull-left" href="#">
      <img class="media-object" src="img/64.jpg"
      alt="媒体对象">
   </a>
   <div class="media-body">
      <h4 class="media-heading">媒体标题</h4>
      这是一些示例文本。这是一些示例文本。
```

```
      这是一些示例文本。这是一些示例文本。
      这是一些示例文本。这是一些示例文本。
      这是一些示例文本。这是一些示例文本。
      这是一些示例文本。这是一些示例文本。
      <div class="media">
         <a class="pull-left" href="#">
            <img class="media-object" src="img/64.jpg"
            alt="媒体对象">
         </a>
         <div class="media-body">
            <h4 class="media-heading">媒体标题</h4>
            这是一些示例文本。这是一些示例文本。
            这是一些示例文本。这是一些示例文本。
            这是一些示例文本。这是一些示例文本。
            这是一些示例文本。这是一些示例文本。
            这是一些示例文本。这是一些示例文本。
         </div>
      </div>
   </div>
</div>

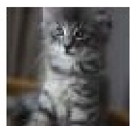
</body>
</html>
```

结果如下所示：



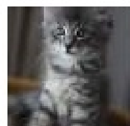
媒体标题

这是一些示例文本。这是一些示例文本。 这是一些示例文本。这是一些示例文本。 这是一些示例文本。这是一些示例文本。 这是一些示例文本。这是一些示例文本。 这是一些示例文本。这是一些示例文本。



媒体标题

这是一些示例文本。这是一些示例文本。 这是一些示例文本。这是一些示例文本。 这是一些示例文本。这是一些示例文本。 这是一些示例文本。这是一些示例文本。 这是一些示例文本。这是一些示例文本。



媒体标题

这是一些示例文本。这是一些示例文本。 这是一些示例文本。这是一些示例文本。 这是一些示例文本。这是一些示例文本。 这是一些示例文本。这是一些示例文本。 这是一些示例文本。这是一些示例文本。

让我们来看看下面这个有关媒体对象列表 .media-list 的实例：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 媒体对象列表</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
```

```
    <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<ul class="media-list">
   <li class="media">
      <a class="pull-left" href="#">
         <img class="media-object" src="img/64.jpg"
         alt="通用的占位符图像">
      </a>
      <div class="media-body">
         <h4 class="media-heading">媒体标题</h4>
         <p>这是一些示例文本。这是一些示例文本。
            这是一些示例文本。这是一些示例文本。
            这是一些示例文本。这是一些示例文本。
            这是一些示例文本。这是一些示例文本。
            这是一些示例文本。这是一些示例文本。</p>
         <!-- 嵌套的媒体对象 -->
         <div class="media">
            <a class="pull-left" href="#">
               <img class="media-object" src="img/64.jpg"
               alt="通用的占位符图像">
            </a>
            <div class="media-body">
               <h4 class="media-heading">嵌套的媒体标题</h4>
               这是一些示例文本。这是一些示例文本。
               这是一些示例文本。这是一些示例文本。
               这是一些示例文本。这是一些示例文本。
               这是一些示例文本。这是一些示例文本。
               这是一些示例文本。这是一些示例文本。
               <!-- 嵌套的媒体对象 -->
               <div class="media">
                  <a class="pull-left" href="#">
                     <img class="media-object" src="img/64.jpg"
                     alt="通用的占位符图像">
                  </a>
                  <div class="media-body">
                     <h4 class="media-heading">嵌套的媒体标题</h4>
                     这是一些示例文本。这是一些示例文本。
                     这是一些示例文本。这是一些示例文本。
                     这是一些示例文本。这是一些示例文本。
                     这是一些示例文本。这是一些示例文本。
                     这是一些示例文本。这是一些示例文本。
                  </div>
               </div>
            </div>
         </div>
         <!-- 嵌套的媒体对象 -->
         <div class="media">
            <a class="pull-left" href="#">
               <img class="media-object" src="img/64.jpg"
               alt="通用的占位符图像">
            </a>
```

```
              <div class="media-body">
                 <h4 class="media-heading">嵌套的媒体标题</h4>
                 这是一些示例文本。这是一些示例文本。
                 这是一些示例文本。这是一些示例文本。
                 这是一些示例文本。这是一些示例文本。
                 这是一些示例文本。这是一些示例文本。
                 这是一些示例文本。这是一些示例文本。
              </div>
           </div>
        </div>
     </li>
     <li class="media">
        <a class="pull-right" href="#">
           <img class="media-object" src="img/64.jpg"
           alt="通用的占位符图像">
        </a>
        <div class="media-body">
           <h4 class="media-heading">媒体标题</h4>
           这是一些示例文本。这是一些示例文本。
           这是一些示例文本。这是一些示例文本。
           这是一些示例文本。这是一些示例文本。
           这是一些示例文本。这是一些示例文本。
           这是一些示例文本。这是一些示例文本。
        </div>
     </li>
  </ul>

  </body>
  </html>
```

结果如下所示：

媒体标题

这是一些示例文本。这是一些示例文本。 这是一些示例文本。这是一些示例文本。 这是一些示例文本。这是一些示例文本。 这是一些示例文本。这是一些示例文本。 这是一些示例文本。这是一些示例文本。

嵌套的媒体标题

这是一些示例文本。这是一些示例文本。 这是一些示例文本。这是一些示例文本。 这是一些示例文本。这是一些示例文本。 这是一些示例文本。这是一些示例文本。 这是一些示例文本。这是一些示例文本。

嵌套的媒体标题

这是一些示例文本。这是一些示例文本。 这是一些示例文本。这是一些示例文本。 这是一些示例文本。这是一些示例文本。 这是一些示例文本。这是一些示例文本。 这是一些示例文本。这是一些示例文本。

嵌套的媒体标题

这是一些示例文本。这是一些示例文本。 这是一些示例文本。这是一些示例文本。 这是一些示例文本。这是一些示例文本。 这是一些示例文本。这是一些示例文本。 这是一些示例文本。这是一些示例文本。

媒体标题

这是一些示例文本。这是一些示例文本。 这是一些示例文本。这是一些示例文本。 这是一些示例文本。这是一些示例文本。 这是一些示例文本。这是一些示例文本。 这是一些示例文本。这是一些示例文本。

Bootstrap 列表组

本章我们将讲解列表组。列表组件用于以列表形式呈现复杂的和自定义的内容。创建一个基本的列表组的步骤如下：

- 向元素 <ul> 添加 class **.list-group**。
- 向 <li> 添加 class **.list-group-item**。

下面的实例演示了这点：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 基本的列表组</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<ul class="list-group">
   <li class="list-group-item">免费域名注册</li>
   <li class="list-group-item">免费 Window 空间托管</li>
   <li class="list-group-item">图像的数量</li>
   <li class="list-group-item">24*7 支持</li>
   <li class="list-group-item">每年更新成本</li>
</ul>

</body>
</html>
```

结果如下所示：

免费域名注册
免费 Window 空间托管
图像的数量
24*7 支持
每年更新成本

向列表组添加徽章

我们可以向任意的列表项添加徽章组件，它会自动定位到右边。只需要在 <li> 元素中添加 **<span class="badge">** 即可。下面的实例演示了这点：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 向列表组添加徽章</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<ul class="list-group">
   <li class="list-group-item">免费域名注册</li>
   <li class="list-group-item">免费 Window 空间托管</li>
   <li class="list-group-item">图像的数量</li>
   <li class="list-group-item">
      <span class="badge">新</span>
      24*7 支持
   </li>
   <li class="list-group-item">每年更新成本</li>
   <li class="list-group-item">
      <span class="badge">新</span>
      折扣优惠
   </li>
</ul>

</body>
</html>
```

结果如下所示：

免费域名注册
免费 Window 空间托管
图像的数量
24*7 支持 新
每年更新成本
折扣优惠 新

向列表组添加链接

通过使用锚标签代替列表项，我们可以向列表组添加链接。我们需要使用 <div> 代替 <ul> 元素。下面的实例演示了这点：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 向列表组添加链接</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<a href="#" class="list-group-item active">
   免费域名注册
</a>
<a href="#" class="list-group-item">24*7 支持</a>
<a href="#" class="list-group-item">免费 Window 空间托管</a>
<a href="#" class="list-group-item">图像的数量</a>
<a href="#" class="list-group-item">每年更新成本</a>

</body>
</html>
```

结果如下所示：

免费域名注册

24*7 支持

免费 Window 空间托管

图像的数量

每年更新成本

向列表组添加自定义内容

我们可以向上面已添加链接的列表组添加任意的 HTML 内容。下面的实例演示了这点：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 向列表组添加自定义内容</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
```

```
    <script src="/scripts/jquery.min.js"></script>
    <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="list-group">
    <a href="#" class="list-group-item active">
        <h4 class="list-group-item-heading">
            入门网站包
        </h4>
    </a>
    <a href="#" class="list-group-item">
        <h4 class="list-group-item-heading">
            免费域名注册
        </h4>
        <p class="list-group-item-text">
            您将通过网页进行免费域名注册。
        </p>
    </a>
    <a href="#" class="list-group-item">
        <h4 class="list-group-item-heading">
            24*7 支持
        </h4>
        <p class="list-group-item-text">
            我们提供 24*7 支持。
        </p>
    </a>
</div>
<div class="list-group">
    <a href="#" class="list-group-item active">
        <h4 class="list-group-item-heading">
            商务网站包
        </h4>
    </a>
    <a href="#" class="list-group-item">
        <h4 class="list-group-item-heading">
            免费域名注册
        </h4>
        <p class="list-group-item-text">
            您将通过网页进行免费域名注册。
        </p>
    </a>
    <a href="#" class="list-group-item">
        <h4 class="list-group-item-heading">24*7 支持</h4>
        <p class="list-group-item-text">我们提供 24*7 支持。</p>
    </a>
</div>

</body>
</html>
```

结果如下所示：

入门网站包

免费域名注册
您将通过网页进行免费域名注册。

24*7 支持
我们提供 24*7 支持。

商务网站包

免费域名注册
您将通过网页进行免费域名注册。

24*7 支持
我们提供 24*7 支持。

Bootstrap 面板（Panels）

本章将讲解 Bootstrap 面板（Panels）。面板组件用于把 DOM 组件插入到一个盒子中。创建一个基本的面板，只需要向 <div> 元素添加 class **.panel** 和 class **.panel-default** 即可，如下面的实例所示：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 默认的面板</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="panel panel-default">
   <div class="panel-body">
      这是一个基本的面板
   </div>
</div>

</body>
</html>
```

结果如下所示：

这是一个基本的面板

面板标题

我们可以通过以下两种方式来添加面板标题：

- 使用 **.panel-heading** class 可以很简单地向面板添加标题容器。to easily add a heading container to your panel.
- 使用带有 **.panel-title** class 的 <h1>-<h6> 来添加预定义样式的标题。

下面的实例演示了这两种方式：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 面板标题</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="panel panel-default">
   <div class="panel-heading">
      不带 title 的面板标题
   </div>
   <div class="panel-body">
      面板内容
   </div>
</div>

<div class="panel panel-default">
   <div class="panel-heading">
      <h3 class="panel-title">
         带有 title 的面板标题
      </h3>
   </div>
   <div class="panel-body">
      面板内容
   </div>
</div>

</body>
</html>
```

结果如下所示：

不带 title 的面板标题

面板内容

带有 title 的面板标题

面板内容

面板脚注

我们可以在面板中添加脚注，只需要把按钮或者副文本放在带有 class **.panel-footer** 的 <div> 中即可。下面的实例演示了这点：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 面板脚注</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="panel panel-default">
   <div class="panel-body">
      这是一个基本的面板
   </div>
   <div class="panel-footer">面板脚注</div>
</div>

</body>
</html>
```

结果如下所示：

这是一个基本的面板

面板脚注

> 面版脚注不会从带语境色彩的面板中继承颜色和边框，因为它不是前景中的内容。

带语境色彩的面板

使用语境状态类 **panel-primary**、**panel-success**、**panel-info**、**panel-warning**、**panel-danger**，来设置带语境色彩的面板，实例如下：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 带语境色彩的面板</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>
```

```
<div class="panel panel-primary">
   <div class="panel-heading">
      <h3 class="panel-title">面板标题</h3>
   </div>
   <div class="panel-body">
      这是一个基本的面板
   </div>
</div>
<div class="panel panel-success">
   <div class="panel-heading">
      <h3 class="panel-title">面板标题</h3>
   </div>
   <div class="panel-body">
      这是一个基本的面板
   </div>
</div>
<div class="panel panel-info">
   <div class="panel-heading">
      <h3 class="panel-title">面板标题</h3>
   </div>
   <div class="panel-body">
      这是一个基本的面板
   </div>
</div>
<div class="panel panel-warning">
   <div class="panel-heading">
      <h3 class="panel-title">面板标题</h3>
   </div>
   <div class="panel-body">
      这是一个基本的面板
   </div>
</div>
<div class="panel panel-danger">
   <div class="panel-heading">
      <h3 class="panel-title">面板标题</h3>
   </div>
   <div class="panel-body">
      这是一个基本的面板
   </div>
</div>

</body>
</html>
```

结果如下所示：

面板标题

这是一个基本的面板

面板标题

这是一个基本的面板

面板标题

这是一个基本的面板

面板标题

这是一个基本的面板

面板标题

这是一个基本的面板

带表格的面板

为了在面板中创建一个无边框的表格，我们可以在面板中使用 class **.table**。假设有个 <div> 包含 **.panel-body**，我们可以向表格的顶部添加额外的边框用来分隔。如果没有包含 **.panel-body** 的 <div>，则组件会无中断地从面板头部移动到表格。

下面的实例演示了这点：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 带表格的面板</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="panel panel-default">
   <div class="panel-heading">
      <h3 class="panel-title">面板标题</h3>
   </div>
   <div class="panel-body">
      这是一个基本的面板
   </div>
   <table class="table">
      <th>产品</th><th>价格 </th>
      <tr><td>产品 A</td><td>200</td></tr>
      <tr><td>产品 B</td><td>400</td></tr>
   </table>
</div>
<div class="panel panel-default">
   <div class="panel-heading">面板标题</div>
   <table class="table">
      <th>产品</th><th>价格 </th>
      <tr><td>产品 A</td><td>200</td></tr>
      <tr><td>产品 B</td><td>400</td></tr>
   </table>
</div>

</body>
</html>
```

结果如下所示：

面板标题

这是一个基本的面板

产品	价格
产品 A	200
产品 B	400

面板标题

产品	价格
产品 A	200
产品 B	400

带列表组的面板

我们可以在任何面板中包含列表组，通过在 <div> 元素中添加 **.panel** 和 **.panel-default** 类来创建面板，并在面板中添加列表组。您可以从 列表组 一章中学习如何创建列表组。

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 带列表组的面板</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="panel panel-default">
   <div class="panel-heading">面板标题</div>
      <div class="panel-body">
         <p>这是一个基本的面板内容。这是一个基本的面板内容。
         这是一个基本的面板内容。这是一个基本的面板内容。
         这是一个基本的面板内容。这是一个基本的面板内容。
         这是一个基本的面板内容。这是一个基本的面板内容。
         </p>
   </div>
   <ul class="list-group">
      <li class="list-group-item">免费域名注册</li>
      <li class="list-group-item">免费 Window 空间托管</li>
      <li class="list-group-item">图像的数量</li>
      <li class="list-group-item">24*7 支持</li>
      <li class="list-group-item">每年更新成本</li>
   </ul>
</div>

</body>
</html>
```

结果如下所示：

面板标题

这是一个基本的面板内容。这是一个基本的面板内容。 这是一个基本的面板内容。这是一个基本的面板内容。 这是一个基本的面板内容。这是一个基本的面板内容。 这是一个基本的面板内容。这是一个基本的面板内容。

- 免费域名注册
- 免费 Window 空间托管
- 图像的数量
- 24*7 支持
- 每年更新成本

Bootstrap Wells

Well 是一种会引起内容凹陷显示或插图效果的容器 <div>。为了创建 Well，只需要简单地把内容放在带有 class **.well** 的 <div> 中即可。下面的实例演示了一个默认的 Well：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 默认的 Well</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

 <div class="well">您好，我在 Well 中！</div>

</body>
</html>
```

结果如下所示：

您好，我在 Well 中！

尺寸大小

您可以使用可选的 class **well-lg** or **well-lg** 来改变 Well 的尺寸大小。这两个类是与 **.well** 类结合使用的。这两个类会影响内边距（padding），根据使用的类，Well 会显示得更大或者更小。

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - Well 的尺寸大小</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="well well-lg">您好，我在大的 Well 中！</div>
<div class="well well-sm">您好，我在小的 Well 中！</div>

</body>
</html>
```

结果如下所示：

您好，我在大的 Well 中!

您好，我在小的 Well 中!

Bootstrap 插件

Bootstrap 插件概览

在前面 布局组件 章节中所讨论到的组件仅仅是个开始。Bootstrap 自带 12 种 jQuery 插件，扩展了功能，可以给站点添加更多的互动。即使您不是一名高级的 JavaScript 开发人员，您也可以着手学习 Bootstrap 的 JavaScript 插件。利用 Bootstrap 数据 API（Bootstrap Data API），大部分的插件可以在不编写任何代码的情况被触发。

站点引用 Bootstrap 插件的方式有两种：

- 单独引用：使用 Bootstrap 的个别的 *.js* 文件。一些插件和 CSS 组件依赖于其他插件。如果您单独引用插件，请先确保弄清这些插件之间的依赖关系。
- 编译（同时）引用：使用 *bootstrap.js* 或压缩版的 *bootstrap.min.js*。

> 不要尝试同时引用这两个文件，因为 *bootstrap.js* 和 *bootstrap.min.js* 都包含了所有的插件。

所有的插件依赖于 ***jQuery***。所以必须在插件文件之前引用 ***jQuery***。请访问 ***bower.json*** 查看 ***Bootstrap*** 当前支持的 ***jQuery*** 版本。

data 属性

- 你可以仅仅通过 data 属性 API 就能使用所有的 Bootstrap 插件，无需写一行 JavaScript 代码。这是 Bootstrap 中的一等 API，也应该是你的首选方式。
- 话又说回来，在某些情况下可能需要将此功能关闭。因此，我们还提供了关闭 data 属性 API 的方法，即解除以 *data-api* 为命名空间并绑定在文档上的事件。就像下面这样：

```
$(document).off('.data-api')
```

- 如需关闭一个特定的插件，只需要在 data-api 命名空间前加上该插件的名称作为命名空间即可，如下所示：

```
$(document).off('.alert.data-api')
```

编程方式的 API

我们为所有 Bootstrap 插件提供了纯 JavaScript 方式的 API。所有公开的 API 都是支持单独或链式调用方式，并且返回其所操作的元素集合（注：和jQuery的调用形式一致）。例如：

```
$(".btn.danger").button("toggle").addClass("fat")
```

所有的方法都可以接受一个可选的选项对象作为参数，或者一个代表特定方法的字符串，或者不带任何参数（这种情况下，将会初始化插件为默认行为），如下所示：

```
// 初始化为默认行为
$("#myModal").modal()
 // 初始化为不支持键盘
$("#myModal").modal({ keyboard: false })
// 初始化并立即调用 show
$("#myModal").modal('show')
```

每个插件在 **Constructor** 属性上也暴露了其原始的构造函数：*$.fn.popover.Constructor*。如果您想获取某个特定插件的实例，可以直接通过页面元素获取：

```
 $('[rel=popover]').data('popover').
```

避免命名空间冲突

某些时候 Bootstrap 插件可能需要与其他 UI 框架一起使用。在这种情况下，可能会发生命名空间冲突。如果不幸发生了这种情况，你可以通过调用插件的 **.noConflict** 方法恢复其原始值。

```
// 返回 $.fn.button 之前所赋的值
var bootstrapButton = $.fn.button.noConflict()
// 为 $().bootstrapBtn 赋予 Bootstrap 功能
$.fn.bootstrapBtn = bootstrapButton
```

事件

Bootstrap 为大多数插件的独特行为提供了自定义事件。一般来说，这些事件有两种形式：

- 动词不定式：这会在事件开始时被触发。例如 *ex: show*。动词不定式事件提供了 *preventDefault* 功能。这使得在事件开始前可以停止操作的执行。

```
$('#myModal').on('show.bs.modal', function (e) {
// 阻止模态框的显示
  if (!data) return e.preventDefault()
})
```

- 过去分词形式：这会在动作执行完毕之后被触发。例如 *ex: shown*。

Bootstrap 过渡效果（Transition）插件

过渡效果（Transition）插件提供了简单的过渡效果。

> 如果您想要单独引用该插件的功能，那么除了其他的 JS 文件，您还需要引用 **transition.js**。或者，正如 Bootstrap 插件概览 一章中所提到，您可以引用 *bootstrap.js* 或压缩版的 *bootstrap.min.js*。

Transition.js 是 transitionEnd 事件和 CSS 过渡效果模拟器的基本帮助器类。它被其他插件用来检查 CSS 过渡效果支持，并用来获取过渡效果。

使用案例

过渡效果（Transition）插件的使用案例：

- 具有幻灯片或淡入效果的模态对话框。具体实例参见 Bootstrap 模态框（Modal）插件。
- 具有淡出效果的标签页。具体实例参见 Bootstrap 标签页（Tab）插件。
- 具有淡出效果的警告框。 具体实例参见 Bootstrap 警告框（Alert）插件。
- 具有幻灯片效果的轮播板。具体实例参见 Bootstrap 轮播（Carousel）插件。

Bootstrap 模态框（Modal）插件

模态框（Modal）是覆盖在父窗体上的子窗体。通常，目的是显示来自一个单独的源的内容，可以在不离开父窗体的情况下有一些互动。子窗体可提供信息、交互等。

> 如果您想要单独引用该插件的功能，那么您需要引用 **modal.js**。或者，正如 Bootstrap 插件概览 一章中所提到，您可以引用 *bootstrap.js* 或压缩版的 *bootstrap.min.js*。

用法

您可以切换模态框（Modal）插件的隐藏内容：

- 通过 **data** 属性：在控制器元素（比如按钮或者链接）上设置属性 **data-toggle="modal"**，同时设置 **data-target="#identifier"** 或 **href="#identifier"** 来指定要切换的特定的模态框（带有 id="identifier"）。
- 通过 **JavaScript**：使用这种技术，您可以通过简单的一行 JavaScript 来调用带有 id="identifier" 的模态框：

```
$('#identifier').modal(options)
```

实例

一个静态的模态窗口实例，如下面的实例所示：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 模态框（Modal）插件</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<h2>创建模态框（Modal）</h2>
<!-- 按钮触发模态框 -->
<button class="btn btn-primary btn-lg" data-toggle="modal"
   data-target="#myModal">
   开始演示模态框
</button>

<!-- 模态框（Modal） -->
<div class="modal fade" id="myModal" tabindex="-1" role="dialog"
   aria-labelledby="myModalLabel" aria-hidden="true">
   <div class="modal-dialog">
      <div class="modal-content">
         <div class="modal-header">
            <button type="button" class="close"
               data-dismiss="modal" aria-hidden="true">
                  &times;
            </button>
            <h4 class="modal-title" id="myModalLabel">
               模态框（Modal）标题
            </h4>
         </div>
         <div class="modal-body">
            在这里添加一些文本
         </div>
         <div class="modal-footer">
            <button type="button" class="btn btn-default"
               data-dismiss="modal">关闭
            </button>
            <button type="button" class="btn btn-primary">
               提交更改
            </button>
         </div>
      </div><!-- /.modal-content -->
</div><!-- /.modal -->

</body>
</html>
```

结果如下所示：



代码讲解：

- 使用模态窗口，您需要有某种触发器。您可以使用按钮或链接。这里我们使用的是按钮。
- 如果您仔细查看上面的代码，您会发现在 <button> 标签中，**data-target="#myModal"** 是您想要在页面上加载的模态框的目标。您可以在页面上创建多个模态框，然后为每个模态框创建不同的触发器。现在，很明显，您不能在同一时间加载多个模块，但您可以在页面上创建多个在不同时间进行加载。
- 在模态框中需要注意两点：
  - 第一是 **.modal**，用来把 <div> 的内容识别为模态框。
  - 第二是 **.fade** class。当模态框被切换时，它会引起内容淡入淡出。
- **aria-labelledby="myModalLabel"**，该属性引用模态框的标题。
- 属性 **aria-hidden="true"** 用于保持模态窗口不可见，直到触发器被触发为止（比如点击在相关的按钮上）。
- <div class="modal-header">，modal-header 是为模态窗口的头部定义样式的类。
- **class="close"**，close 是一个 CSS class，用于为模态窗口的关闭按钮设置样式。
- **data-dismiss="modal"**，是一个自定义的 HTML5 data 属性。在这里它被用于关闭模态窗口。
- **class="modal-body"**，是 Bootstrap CSS 的一个 CSS class，用于为模态窗口的主体设置样式。
- **class="modal-footer"**，是 Bootstrap CSS 的一个 CSS class，用于为模态窗口的底部设置样式。
- **data-toggle="modal"**，HTML5 自定义的 data 属性 data-toggle 用于打开模态窗口。

选项

有一些选项可以用来定制模态窗口（Modal Window）的外观和感观，它们是通过 data 属性或 JavaScript 来传递的。下表列出了这些选项：

选项名称	类型/默认值	Data 属性名称	
backdrop	boolean 或 string 'static' 默认值：*true*	data-backdrop	指定一个静态的背景，当用户点击模态框外
keyboard	boolean 默认值：*true*	data-keyboard	当按下 escape 键时关闭模态框，设置为 fals
show	boolean 默认值：*true*	data-show	当初始化时显示模态框。
remote	path 默认值：*false*	data-remote	使用 jQuery *.load* 方法，为模态框的主体注 容。如下面的实例所 示： `<a data-toggle="modal" href='`

方法

下面是一些可与 modal() 一起使用的有用的方法。

方法	描述	实例
Options: .modal(options)	把内容作为模态框激活。接受一个可选的选项对象。	`$('#identifier').modal({keyboard: false})`
Toggle: .modal('toggle')	手动切换模态框。	`$('#identifier').modal('toggle')`
Show: .modal('show')	手动打开模态框。	`$('#identifier').modal('show')`
Hide: .modal('hide')	手动隐藏模态框。	`$('#identifier').modal('hide')`

实例

下面的实例演示了方法的用法：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 模态框（Modal）插件方法</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<h2>模态框（Modal）插件方法</h2>

<!-- 按钮触发模态框 -->
<button class="btn btn-primary btn-lg" data-toggle="modal" data-ta
   开始演示模态框
```

```
</button>

<!-- 模态框（Modal） -->
<div class="modal fade" id="myModal" tabindex="-1" role="dialog"
   aria-labelledby="myModalLabel" aria-hidden="true">
   <div class="modal-dialog">
      <div class="modal-content">
         <div class="modal-header">
            <button type="button" class="close" data-dismiss="modal
               aria-hidden="true">×
            </button>
            <h4 class="modal-title" id="myModalLabel">
               模态框（Modal）标题
            </h4>
         </div>
         <div class="modal-body">
            按下 ESC 按钮退出。
         </div>
         <div class="modal-footer">
            <button type="button" class="btn btn-default"
               data-dismiss="modal">关闭
            </button>
            <button type="button" class="btn btn-primary">
               提交更改
            </button>
         </div>
      </div><!-- /.modal-content -->
   </div><!-- /.modal-dialog -->
</div><!-- /.modal -->
<script>
   $(function () { $('#myModal').modal({
      keyboard: true
   })});
</script>

</body>
</html>
```

结果如下所示：



只需要点击 ESC 键，模态窗口即会退出。

事件

下表列出了模态框中要用到事件。这些事件可在函数中当钩子使用。

事件	描述	实例
show.bs.modal	在调用 show 方法后触发。	`$('#identifier').on('show.bs.modal', func`
shown.bs.modal	当模态框对用户可见时触发（将等待 CSS 过渡效果完成）。	`$('#identifier').on('shown.bs.modal', fun`
hide.bs.modal	当调用 hide 实例方法时触发。	`$('#identifier').on('hide.bs.modal', func`
hidden.bs.modal	当模态框完全对用户隐藏时触发。	`$('#identifier').on('hidden.bs.modal', fu`

实例

下面的实例演示了事件的用法：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 模态框（Modal）插件事件</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<h2>模态框（Modal）插件事件</h2>
```

```
<!-- 按钮触发模态框 -->
<button class="btn btn-primary btn-lg" data-toggle="modal" data-tar
   开始演示模态框
</button>

<!-- 模态框（Modal） -->
<div class="modal fade" id="myModal" tabindex="-1" role="dialog"
   aria-labelledby="myModalLabel" aria-hidden="true">
   <div class="modal-dialog">
      <div class="modal-content">
         <div class="modal-header">
            <button type="button" class="close" data-dismiss="modal
               aria-hidden="true">×
            </button>
            <h4 class="modal-title" id="myModalLabel">
               模态框（Modal）标题
            </h4>
         </div>
         <div class="modal-body">
            点击关闭按钮检查事件功能。
         </div>
         <div class="modal-footer">
            <button type="button" class="btn btn-default"
               data-dismiss="modal">
               关闭
            </button>
            <button type="button" class="btn btn-primary">
               提交更改
            </button>
         </div>
      </div><!-- /.modal-content -->
   </div><!-- /.modal-dialog -->
</div><!-- /.modal -->
<script>
   $(function () { $('#myModal').modal('hide')})});
</script>
<script>
   $(function () { $('#myModal').on('hide.bs.modal', function () {
      alert('嘿，我听说您喜欢模态框...');})
   });
</script>

</body>
</html>
```

结果如下所示：



正如上面实例所示，如果您点击了 关闭 按钮，即 *hide* 事件，则会显示一个警告消息。

Bootstrap 下拉菜单（Dropdown）插件

Bootstrap 下拉菜单 这一章讲解了下拉菜单，但是没有涉及到交互部分，本章将具体讲解下拉菜单的交互。使用下拉菜单（Dropdown）插件，您可以向任何组件（比如导航栏、标签页、胶囊式导航菜单、按钮等）添加下拉菜单。

如果您想要单独引用该插件的功能，那么您需要引用 **dropdown.js**。或者，正如 Bootstrap 插件概览 一章中所提到，您可以引用 *bootstrap.js* 或压缩版的 *bootstrap.min.js*。

用法

您可以切换下拉菜单（Dropdown）插件的隐藏内容：

- 通过 **data** 属性：向链接或按钮添加 **data-toggle="dropdown"** 来切换下拉菜单，如下所示：

```
&lt;div class="dropdown"&gt;
  &lt;a data-toggle="dropdown" href="#"&gt;下拉菜单（Dropdown）触
  &lt;ul class="dropdown-menu" role="menu" aria-labelledby="dLa
    ...
  &lt;/ul&gt;
&lt;/div&gt;
```

  如果您需要保持链接完整（在浏览器不启用 JavaScript 时有用），请使用 **data-target** 属性代替 **href="#"**：

```
&lt;div class="dropdown"&gt;
  &lt;a id="dLabel" role="button" data-toggle="dropdown" data-t
    下拉菜单（Dropdown） &lt;span class="caret"&gt;&lt;/span&gt;
  &lt;/a&gt;

  &lt;ul class="dropdown-menu" role="menu" aria-labelledby="dLa
    ...
  &lt;/ul&gt;
&lt;/div&gt;
```

- 通过 **JavaScript**：通过 JavaScript 调用下拉菜单切换，请使用下面的方法：

```
$('.dropdown-toggle').dropdown()
```

实例

在导航栏内

下面的实例演示了在导航栏内的下拉菜单的用法：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 默认的导航栏</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<nav class="navbar navbar-default" role="navigation">
   <div class="navbar-header">
      <a class="navbar-brand" href="#">W3Cschool</a>
   </div>
   <div>
      <ul class="nav navbar-nav">
         <li class="active"><a href="#">iOS</a></li>
         <li><a href="#">SVN</a></li>
         <li class="dropdown">
            <a href="#" class="dropdown-toggle" data-toggle="dropdo
               Java
               <b class="caret"></b>
            </a>
            <ul class="dropdown-menu">
               <li><a href="#">jmeter</a></li>
               <li><a href="#">EJB</a></li>
               <li><a href="#">Jasper Report</a></li>
               <li class="divider"></li>
               <li><a href="#">分离的链接</a></li>
               <li class="divider"></li>
               <li><a href="#">另一个分离的链接</a></li>
            </ul>
         </li>
      </ul>
   </div>
</nav>

</body>
</html>
```

结果如下所示：

W3Cschool iOS SVN Java

在标签页内

下面的实例演示了在标签页内的下拉菜单的用法：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 带有下拉菜单的标签页</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<p>带有下拉菜单的标签页</p>
<ul class="nav nav-tabs">
   <li class="active"><a href="#">Home</a></li>
   <li><a href="#">SVN</a></li>
   <li><a href="#">iOS</a></li>
   <li><a href="#">VB.Net</a></li>
   <li class="dropdown">
      <a class="dropdown-toggle" data-toggle="dropdown" href="#">
         Java <span class="caret"></span>
      </a>
      <ul class="dropdown-menu">
         <li><a href="#">Swing</a></li>
         <li><a href="#">jMeter</a></li>
         <li><a href="#">EJB</a></li>
         <li class="divider"></li>
         <li><a href="#">分离的链接</a></li>
      </ul>
   </li>
   <li><a href="#">PHP</a></li>
</ul>

</body>
</html>
```

结果如下所示：



选项

没有选项。

方法

下拉菜单切换有一个简单的方法用来显示或隐藏下拉菜单。

```
$().dropdown('toggle')
```

实例

下面的实例演示了下拉菜单（Dropdown）插件方法的用法：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 下拉菜单（Dropdown）插件方法</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<nav class="navbar navbar-default" role="navigation">
   <div class="navbar-header">
      <a class="navbar-brand" href="#">W3Cschool</a>
   </div>

   <div id="myexample">
      <ul class="nav navbar-nav">
         <li class="active"><a href="#">iOS</a></li>
         <li><a href="#">SVN</a></li>
         <li class="dropdown">
            <a href="#" class="dropdown-toggle">Java <b
               class="caret"></b></a>
               <ul class="dropdown-menu">
                  <li><a id="action-1" href="#">
                     jmeter</a>
                  </li>
                  <li><a href="#">EJB</a></li>
                  <li><a href="#">Jasper Report</a></li>
                  <li class="divider"></li>
                  <li><a href="#">分离的链接</a></li>
                  <li class="divider"></li>
                  <li><a href="#">另一个分离的链接</a></li>
               </ul>
            </li>
         </ul>
      </div>
   </nav>
</div>
<script>
   $(function(){
      $(".dropdown-toggle").dropdown('toggle');
      });
</script>

</body>
</html>
```

结果如下所示：


W3Cschool
iOS
SVN
Java
jmeter
EJB
Jasper Report
分离的链接
另一个分离的链接

Bootstrap 滚动监听（Scrollspy）插件

滚动监听（Scrollspy）插件，即自动更新导航插件，会根据滚动条的位置自动更新对应的导航目标。其基本的实现是随着您的滚动，基于滚动条的位置向导航栏添加 **.active** class。

如果您想要单独引用该插件的功能，那么您需要引用 **scrollspy.js**。或者，正如 Bootstrap 插件概览 一章中所提到，您可以引用 *bootstrap.js* 或压缩版的 *bootstrap.min.js*。

用法

您可以向顶部导航添加滚动监听行为：

- 通过 **data** 属性：向您想要监听的元素（通常是 body）添加 **data-spy="scroll"**。然后添加带有 Bootstrap **.nav** 组件的父元素的 ID 或 class 的属性 **data-target**。为了它能正常工作，您必须确保页面主体中有匹配您所要监听链接的 ID 的元素存在。

```
&lt;body data-spy="scroll" data-target=".navbar-example"&gt;
  ...
  &lt;div class="navbar-example"&gt;
    &lt;ul class="nav nav-tabs"&gt;
      ...
    &lt;/ul&gt;
  &lt;/div&gt;
  ...
&lt;/body&gt;
```

- 通过 **JavaScript**：您可以通过 JavaScript 调用滚动监听，选取要监听的元素，然后调用 **.scrollspy()** 函数：

```
$('body').scrollspy({ target: '.navbar-example' })
```

实例

下面的实例演示了通过 data 属性使用滚动监听（Scrollspy）插件：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 滚动监听（Scrollspy）插件</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
```

```
    <script src="/scripts/jquery.min.js"></script>
    <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<nav id="navbar-example" class="navbar navbar-default navbar-static
    <div class="navbar-header">
        <button class="navbar-toggle" type="button" data-toggle="coll
            data-target=".bs-js-navbar-scrollspy">
            <span class="sr-only">切换导航</span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
        </button>
        <a class="navbar-brand" href="#">教程名称</a>
    </div>
    <div class="collapse navbar-collapse bs-js-navbar-scrollspy">
        <ul class="nav navbar-nav">
            <li><a href="#ios">iOS</a></li>
            <li><a href="#svn">SVN</a></li>
            <li class="dropdown">
                <a href="#" id="navbarDrop1" class="dropdown-toggle"
                    data-toggle="dropdown">Java
                    <b class="caret"></b>
                </a>
                <ul class="dropdown-menu" role="menu"
                    aria-labelledby="navbarDrop1">
                    <li><a href="#jmeter" tabindex="-1">jmeter</a></li>
                    <li><a href="#ejb" tabindex="-1">ejb</a></li>
                    <li class="divider"></li>
                    <li><a href="#spring" tabindex="-1">spring</a></li>
                </ul>
            </li>
        </ul>
    </div>
</nav>
<div data-spy="scroll" data-target="#navbar-example" data-offset="0
    style="height:200px;overflow:auto; position: relative;">
    <h4 id="ios">iOS</h4>
    <p>iOS 是一个由苹果公司开发和发布的手机操作系统。最初是于 2007 年首次发布
        TV。iOS 派生自 OS X，它们共享 Darwin 基础。OS X 操作系统是用在苹果电
    </p>
    <h4 id="svn">SVN</h4>
    <p>Apache Subversion，通常缩写为 SVN，是一款开源的版本控制系统软件。Sub
    </p>
    <h4 id="jmeter">jMeter</h4>
    <p>jMeter 是一款开源的测试软件。它是 100% 纯 Java 应用程序，用于负载和性
    </p>
    <h4 id="ejb">EJB</h4>
    <p>Enterprise Java Beans（EJB）是一个创建高度可扩展性和强大企业级应用程
    </p>
    <h4 id="spring">Spring</h4>
    <p>Spring 框架是一个开源的 Java 平台，为快速开发功能强大的 Java 应用程序
```

```
    </p>
    <p>Spring 框架最初是由 Rod Johnson 编写的，在 2003 年 6 月首次发布于 A
    </p>
</div>

</body>
</html>
```

结果如下所示：



选项

通过 data 属性或 JavaScript 来传递。下表列出了这些选项：

选项名称	类型/默认值	Data 属性名称	描述
offset	number 默认值：*10*	data-offset	当计算滚动位置时，距离顶部的偏移像素。

方法

.scrollspy('refresh')：当通过 JavaScript 调用 scrollspy 方法时，您需要调用 **.refresh** 方法来更新 DOM。这在 DOM 的任意元素发生变更（即，您添加或移除了某些元素）时非常有用。下面是使用该方法的语法。

```
$('[data-spy="scroll"]').each(function () {
  var $spy = $(this).scrollspy('refresh')
})
```

实例

下面的实例演示了 **.scrollspy('refresh')** 方法的用法：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 滚动监听（Scrollspy）插件方法</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<nav id="myScrollspy" class="navbar navbar-default navbar-static" r
   <div class="navbar-header">
      <button class="navbar-toggle" type="button" data-toggle="col
         data-target=".bs-js-navbar-scrollspy">
         <span class="sr-only">切换导航</span>
         <span class="icon-bar"></span>
         <span class="icon-bar"></span>
         <span class="icon-bar"></span>
      </button>
      <a class="navbar-brand" href="#">教程名称</a>
   </div>
   <div class="collapse navbar-collapse bs-js-navbar-scrollspy">
      <ul class="nav navbar-nav">
         <li class="active"><a href="#ios">iOS</a></li>
         <li><a href="#svn">SVN</a></li>
         <li class="dropdown">
            <a href="#" id="navbarDrop1" class="dropdown-toggle"
               data-toggle="dropdown">Java
               <b class="caret"></b>
            </a>
            <ul class="dropdown-menu" role="menu"
               aria-labelledby="navbarDrop1">
               <li><a href="#jmeter" tabindex="-1">jmeter</a></li>
               <li><a href="#ejb" tabindex="-1">ejb</a></li>
               <li class="divider"></li>
               <li><a href="#spring" tabindex="-1">spring</a></li>
            </ul>
         </li>
      </ul>
   </div>
</nav>
<div data-spy="scroll" data-target="#myScrollspy" data-offset="0"
   style="height:200px;overflow:auto; position: relative;">
   <div class="section">
      <h4 id="ios">iOS<small><a href="#" onclick="removeSection(thi
         &times; 删除该部分</a></small>
      </h4>
      <p>iOS 是一个由苹果公司开发和发布的手机操作系统。最初是于 2007 年首次发
```

```
            TV。iOS 派生自 OS X，它们共享 Darwin 基础。OS X 操作系统是用在苹果电
        </div>
        <div class="section">
            <h4 id="svn">SVN<small></small></h4>
            <p>Apache Subversion，通常缩写为 SVN，是一款开源的版本控制系统软件。
        </div>
        <div class="section">
            <h4 id="jmeter">jMeter<small><a href="#" onclick="removeSecti
            &times; 删除该部分</a></small>
            </h4>
            <p>jMeter 是一款开源的测试软件。它是 100% 纯 Java 应用程序，用于负载
        </div>
        <div class="section">
            <h4 id="ejb">EJB</h4>
            <p>Enterprise Java Beans（EJB）是一个创建高度可扩展性和强大企业级应
        </div>
        <div class="section">
            <h4 id="spring">Spring</h4>
            <p>Spring 框架是一个开源的 Java 平台，为快速开发功能强大的 Java 应用
            <p>Spring 框架最初是由 Rod Johnson 编写的，在 2003 年 6 月首次发布
        </div>
    </div>
    <script type="text/javascript">
        $(function(){
        removeSection = function(e) {
            $(e).parents(".section").remove();
            $('[data-spy="scroll"]').each(function () {
                var $spy = $(this).scrollspy('refresh')
            });
        }
        $("#myScrollspy").scrollspy();
    });
    </script>

    </body>
    </html>
```

结果如下所示：

教程名称

iOS × 删除该部分

iOS 是一个由苹果公司开发和发布的手机操作系统。最初是于 2007 年首次发布 iPhone、iPod Touch 和 Apple TV。iOS 派生自 OS X，它们共享 Darwin 基础。OS X 操作系统是用在苹果电脑上，iOS 是苹果的移动版本。

SVN

Apache Subversion，通常缩写为 SVN，是一款开源的版本控制系统软件。Subversion 由 CollabNet 公司在 2000 年创建。但是现在它已经发展为 Apache Software Foundation 的一个项目，因此拥有丰富的开发人员和用户社区。

事件

下表列出了滚动监听中要用到事件。这些事件可在函数中当钩子使用。

事件	描述	实
activate.bs.scrollspy	每当一个新项目被滚动监听激活时，触发该事件。	`$('#myScrollspy').on('activate.bs.scroll`

实例

下面的实例演示了 **activate.bs.scrollspy** 事件的用法：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 滚动监听（Scrollspy）插件事件</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>
```

```
<nav id="myScrollspy" class="navbar navbar-default navbar-static" r
   <div class="navbar-header">
      <button class="navbar-toggle" type="button" data-toggle="coll
         data-target=".bs-js-navbar-scrollspy">
         <span class="sr-only">切换导航</span>
         <span class="icon-bar"></span>
         <span class="icon-bar"></span>
         <span class="icon-bar"></span>
      </button>
      <a class="navbar-brand" href="#">教程名称</a>
   </div>
   <div class="collapse navbar-collapse bs-js-navbar-scrollspy">
      <ul class="nav navbar-nav">
         <li class="active"><a href="#ios">iOS</a></li>
         <li><a href="#svn">SVN</a></li>
         <li class="dropdown">
            <a href="#" id="navbarDrop1" class="dropdown-toggle"
               data-toggle="dropdown">
               Java <b class="caret"></b>
            </a>
            <ul class="dropdown-menu" role="menu"
               aria-labelledby="navbarDrop1">
               <li><a href="#jmeter" tabindex="-1">jmeter</a></li>
               <li><a href="#ejb" tabindex="-1">ejb</a></li>
               <li class="divider"></li>
               <li><a href="#spring" tabindex="-1">spring</a></li>
            </ul>
         </li>
      </ul>
   </div>
</nav>
<div data-spy="scroll" data-target="#myScrollspy" data-offset="0"
   style="height:200px;overflow:auto; position: relative;">
   <div class="section">
      <h4 id="ios">iOS<small><a href="#" onclick="removeSection(thi
         &times; 删除该部分</a></small>
      </h4>
      <p>iOS 是一个由苹果公司开发和发布的手机操作系统。最初是于 2007 年首次发
      TV。iOS 派生自 OS X，它们共享 Darwin 基础。OS X 操作系统是用在苹果电
   </div>
   <div class="section">
      <h4 id="svn">SVN<small></small></h4>
      <p>Apache Subversion，通常缩写为 SVN，是一款开源的版本控制系统软件。
   </div>
   <div class="section">
      <h4 id="jmeter">jMeter<small><a href="#" onclick="removeSecti
         &times; 删除该部分</a></small>
      </h4>
      <p>jMeter 是一款开源的测试软件。它是 100% 纯 Java 应用程序，用于负载
   </div>
   <div class="section">
      <h4 id="ejb">EJB</h4>
      <p>Enterprise Java Beans（EJB）是一个创建高度可扩展性和强大企业级应
```

```
      </div>
      <div class="section">
         <h4 id="spring">Spring</h4>
         <p>Spring 框架是一个开源的 Java 平台，为快速开发功能强大的 Java 应用
         <p>Spring 框架最初是由 Rod Johnson 编写的，在 2003 年 6 月首次发布
      </div>
   </div>
   <script type="text/javascript">
      $(function(){
         removeSection = function(e) {
         $(e).parents(".section").remove();
         $('[data-spy="scroll"]').each(function () {
             var $spy = $(this).scrollspy('refresh')
         });
      }
      $("#myScrollspy").scrollspy();
      $('#myScrollspy').on('activate.bs.scrollspy', function () {
         var currentItem = $(".nav li.active > a").text();
         $("#activeitem").html("目前您正在查看 - " + currentItem);
      })
   });
   </script>

   </body>
   </html>
```

结果如下所示：

教程名称

iOS × 删除该部分

iOS 是一个由苹果公司开发和发布的手机操作系统。最初是于 2007 年首次发布 iPhone、iPod Touch 和 Apple TV。iOS 派生自 OS X，它们共享 Darwin 基础。OS X 操作系统是用在苹果电脑上，iOS 是苹果的移动版本。

SVN

Apache Subversion，通常缩写为 SVN，是一款开源的版本控制系统软件。Subversion 由 CollabNet 公司在 2000 年创建。但是现在它已经发展为 Apache Software Foundation 的一个项目，因此拥有丰富的开发人员和用户社区。

Bootstrap 标签页（Tab）插件

标签页（Tab）在 Bootstrap 导航元素 一章中介绍过。通过结合一些 data 属性，您可以轻松地创建一个标签页界面。通过这个插件您可以把内容放置在标签页或者是胶囊式标签页甚至是下拉菜单标签页中。

> 如果您想要单独引用该插件的功能，那么您需要引用 **tab.js**。或者，正如 Bootstrap 插件概览 一章中所提到，您可以引用 *bootstrap.js* 或压缩版的 *bootstrap.min.js*。

用法

您可以通过以下两种方式启用标签页：

- 通过 **data 属性**：您需要添加 **data-toggle="tab"** 或 **data-toggle="pill"** 到锚文本链接中。

  添加 **nav** 和 **nav-tabs** 类到 **ul** 中，将会应用 Bootstrap 标签样式，添加 **nav** 和 **nav-pills** 类到 **ul** 中，将会应用 Bootstrap 胶囊式样式。

  ```
  &lt;ul class="nav nav-tabs"&gt;
     &lt;li&gt;&lt;a href="#identifier" data-toggle="tab"&gt;Home
  ...
  &lt;/ul&gt;
  ```

- 通过 **JavaScript**：您可以使用 Javscript 来启用标签页，如下所示：

  ```
  $('#myTab a').click(function (e) {
    e.preventDefault()
    $(this).tab('show')
  })
  ```

  下面的实例演示了以不同的方式来激活各个标签页：

```
// 通过名称选取标签页
$('#myTab a[href="#profile"]').tab('show')

// 选取第一个标签页
$('#myTab a:first').tab('show')

// 选取最后一个标签页
$('#myTab a:last').tab('show')

// 选取第三个标签页（从 0 开始索引）
$('#myTab li:eq(2) a').tab('show')
```

淡入淡出效果

如果需要为标签页设置淡入淡出效果，请添加 **.fade** 到每个 **.tab-pane** 后面。第一个标签页必须添加 **.in** 类，以便淡入显示初始内容，如下面实例所示：

```
<div class="tab-content">
  <div class="tab-pane fade in active" id="home">...</div>
  <div class="tab-pane fade" id="svn">...</div>
  <div class="tab-pane fade" id="ios">...</div>
  <div class="tab-pane fade" id="java">...</div>
</div>
```

实例

下面的实例演示了使用 data 属性的标签页（Tab）插件及其淡入淡出的效果：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 标签页（Tab）插件</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<ul id="myTab" class="nav nav-tabs">
   <li class="active">
      <a href="#home" data-toggle="tab">
         W3Cschool Home
      </a>
   </li>
   <li><a href="#ios" data-toggle="tab">iOS</a></li>
   <li class="dropdown">
      <a href="#" id="myTabDrop1" class="dropdown-toggle"
         data-toggle="dropdown">Java
         <b class="caret"></b>
      </a>
      <ul class="dropdown-menu" role="menu" aria-labelledby="myTabD
         <li><a href="#jmeter" tabindex="-1" data-toggle="tab">jmet
         <li><a href="#ejb" tabindex="-1" data-toggle="tab">ejb</a>
      </ul>
   </li>
</ul>
<div id="myTabContent" class="tab-content">
   <div class="tab-pane fade in active" id="home">
      <p>W3Cschoool菜鸟教程是一个提供最新的web技术站点，本站免费提供了建站相
   </div>
   <div class="tab-pane fade" id="ios">
      <p>iOS 是一个由苹果公司开发和发布的手机操作系统。最初是于 2007 年首次发
      TV。iOS 派生自 OS X，它们共享 Darwin 基础。OS X 操作系统是用在苹果电
   </div>
   <div class="tab-pane fade" id="jmeter">
      <p>jMeter 是一款开源的测试软件。它是 100% 纯 Java 应用程序，用于负载
   </div>
   <div class="tab-pane fade" id="ejb">
      <p>Enterprise Java Beans（EJB）是一个创建高度可扩展性和强大企业级应
      </p>
   </div>
</div>

</body>
</html>
```

结果如下所示：

W3Cschool Home iOS Java

W3Cschoool菜鸟教程是一个提供最新的web技术站点，本站免费提供了建站相关的技术文档，帮助广大web技术爱好者快速入门并建立自己的网站。菜鸟先飞早入行——学的不仅是技术，更是梦想。

方法

.$().tab：该方法可以激活标签页元素和内容容器。标签页需要用一个 **data-target** 或者一个指向 DOM 中容器节点的 **href**。

```
<ul class="nav nav-tabs" id="myTab">
  <li class="active"><a href="#identifier" data-toggle="tab">Home</
  .....
</ul>
<div class="tab-content">
  <div class="tab-pane active" id="home">...</div>
  .....
</div>
<script>
  $(function () {
    $('#myTab a:last').tab('show')
  })
</script>
```

实例

下面的实例演示了标签页（Tab）插件方法 **.tab** 的用法。在本实例中，第二个标签页 *iOS* 是激活的：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 标签页（Tab）插件方法</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<ul id="myTab" class="nav nav-tabs">
   <li class="active"><a href="#home" data-toggle="tab">
      W3Cschool Home</a>
   </li>
   <li><a href="#ios" data-toggle="tab">iOS</a></li>
   <li class="dropdown">
      <a href="#" id="myTabDrop1" class="dropdown-toggle"
```

```
            data-toggle="dropdown">Java <b class="caret"></b>
         </a>
         <ul class="dropdown-menu" role="menu" aria-labelledby="myTabD
            <li><a href="#jmeter" tabindex="-1" data-toggle="tab">
               jmeter</a>
            </li>
            <li><a href="#ejb" tabindex="-1" data-toggle="tab">
               ejb</a>
            </li>
         </ul>
      </li>
   </ul>
   <div id="myTabContent" class="tab-content">
      <div class="tab-pane fade in active" id="home">
         <p>W3Cschoool菜鸟教程是一个提供最新的web技术站点，本站免费提供了建站相
      </div>
      <div class="tab-pane fade" id="ios">
         <p>iOS 是一个由苹果公司开发和发布的手机操作系统。最初是于 2007 年首次发
         TV。iOS 派生自 OS X，它们共享 Darwin 基础。OS X 操作系统是用在苹果电
      </div>
      <div class="tab-pane fade" id="jmeter">
         <p>jMeter 是一款开源的测试软件。它是 100% 纯 Java 应用程序，用于负载
      </div>
      <div class="tab-pane fade" id="ejb">
         <p>Enterprise Java Beans（EJB）是一个创建高度可扩展性和强大企业级应
         </p>
      </div>
   </div>
   <script>
      $(function () {
         $('#myTab li:eq(1) a').tab('show');
      });
   </script>

   </body>
   </html>
```

结果如下所示：



事件

下表列出了标签页（Tab）插件中要用到的事件。这些事件可在函数中当钩子使用。

事件	描述	
show.bs.tab	该事件在标签页显示时触发，但是必须在新标签页被显示之前。分别使用 **event.target** 和 **event.relatedTarget** 来定位到激活的标签页和前一个激活的标签页。	`$('a[data-toggle="tab"]').on('sh`
shown.bs.tab	该事件在标签页显示时触发，但是必须在某个标签页已经显示之后。分别使用 **event.target** 和 **event.relatedTarget** 来定位到激活的标签页和前一个激活的标签页。	`$('a[data-toggle="tab"]').on('sh`

实例

下面的实例演示了标签页（Tab）插件事件的用法。在本实例中，将显示当前和前一个访问过的标签页：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 标签页（Tab）插件事件</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<hr>
   <p class="active-tab"><strong>激活的标签页</strong>：<span></span>
   <p class="previous-tab"><strong>前一个激活的标签页</strong>：<span>
<hr>
<ul id="myTab" class="nav nav-tabs">
   <li class="active"><a href="#home" data-toggle="tab">
      W3Cschool Home</a></li>
   <li><a href="#ios" data-toggle="tab">iOS</a></li>
   <li class="dropdown">
```

```
        <a href="#" id="myTabDrop1" class="dropdown-toggle"
           data-toggle="dropdown">
           Java <b class="caret"></b></a>
        <ul class="dropdown-menu" role="menu" aria-labelledby="myTabD
           <li><a href="#jmeter" tabindex="-1" data-toggle="tab">jmet
           <li><a href="#ejb" tabindex="-1" data-toggle="tab">ejb</a>
        </ul>
     </li>
  </ul>
  <div id="myTabContent" class="tab-content">
     <div class="tab-pane fade in active" id="home">
        <p>W3Cschoool菜鸟教程是一个提供最新的web技术站点，本站免费提供了建站相
     </div>
     <div class="tab-pane fade" id="ios">
        <p>iOS 是一个由苹果公司开发和发布的手机操作系统。最初是于 2007 年首次发
        TV。iOS 派生自 OS X，它们共享 Darwin 基础。OS X 操作系统是用在苹果电
     </div>
     <div class="tab-pane fade" id="jmeter">
        <p>jMeter 是一款开源的测试软件。它是 100% 纯 Java 应用程序，用于负载
     </div>
     <div class="tab-pane fade" id="ejb">
        <p>Enterprise Java Beans（EJB）是一个创建高度可扩展性和强大企业级应
        </p>
     </div>
  </div>
  <script>
     $(function(){
        $('a[data-toggle="tab"]').on('shown.bs.tab', function (e) {
        // 获取已激活的标签页的名称
        var activeTab = $(e.target).text();
        // 获取前一个激活的标签页的名称
        var previousTab = $(e.relatedTarget).text();
        $(".active-tab span").html(activeTab);
        $(".previous-tab span").html(previousTab);
     });
  });
  </script>

  </body>
  </html>
```

结果如下所示：

激活的标签页： iOS

前一个激活的标签页： W3Cschool Home

W3Cschool Home | iOS | Java ▾

iOS 是一个由苹果公司开发和发布的手机操作系统。最初是于 2007 年首次发布 iPhone、iPod Touch 和 Apple TV。iOS 派生自 OS X，它们共享 Darwin 基础。OS X 操作系统是用在苹果电脑上，iOS 是苹果的移动版本。

Bootstrap 工具提示（Tooltip）插件

当您想要描述一个链接的时候，工具提示（Tooltip）就显得非常有用。工具提示（Tooltip）插件是受 *Jason Frame* 写的 *jQuery.tipsy* 的启发。工具提示（Tooltip）插件做了很多改进，例如不需要依赖图像，而是改用 CSS 实现动画效果，用 data 属性存储标题信息。

> 如果您想要单独引用该插件的功能，那么您需要引用 **tooltip.js**。或者，正如 Bootstrap 插件概览 一章中所提到，您可以引用 *bootstrap.js* 或压缩版的 *bootstrap.min.js*。

用法

工具提示（Tooltip）插件根据需求生成内容和标记，默认情况下是把工具提示（tooltip）放在它们的触发元素后面。您可以有以下两种方式添加工具提示（tooltip）：

- 通过 **data** 属性：如需添加一个工具提示（tooltip），只需向一个锚标签添加 **data-toggle="tooltip"** 即可。锚的 title 即为工具提示（tooltip）的文本。默认情况下，插件把工具提示（tooltip）设置在顶部。

  ```
  &lt;a href="#" data-toggle="tooltip" title="Example tooltip"&gt
  ```

- 通过 **JavaScript**：通过 JavaScript 触发工具提示（tooltip）：

  ```
  $('#identifier').tooltip(options)
  ```

> 工具提示（Tooltip）插件不像之前所讨论的下拉菜单及其他插件那样，它不是纯 CSS 插件。如需使用该插件，您必须使用 jquery 激活它（读取 javascript）。使用下面的脚本来启用页面中的所有的工具提示（tooltip）：
>
> ```
> $(function () { $("[data-toggle='tooltip']").tooltip(); });
> ```

实例

下面的实例演示了通过 data 属性使用工具提示（Tooltip）插件的用法。

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 工具提示（Tooltip）插件</title>
```

```
    <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
    <script src="/scripts/jquery.min.js"></script>
    <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<h4>工具提示（Tooltip）插件 - 锚</h4>
这是一个 <a href="#" class="tooltip-test" data-toggle="tooltip"
    title="默认的 Tooltip">
    默认的 Tooltip
</a>.
这是一个 <a href="#" class="tooltip-test" data-toggle="tooltip"
    data-placement="left" title="左侧的 Tooltip">
    左侧的 Tooltip
</a>.
这是一个 <a href="#" data-toggle="tooltip" data-placement="top"
    title="顶部的 Tooltip">
    顶部的 Tooltip
</a>.
这是一个 <a href="#" data-toggle="tooltip" data-placement="bottom"
    title="底部的 Tooltip">
    底部的 Tooltip
</a>.
这是一个 <a href="#" data-toggle="tooltip" data-placement="right"
    title="右侧的 Tooltip">
    右侧的 Tooltip
</a>

<br>

<h4>工具提示（Tooltip）插件 - 按钮</h4>
<button type="button" class="btn btn-default" data-toggle="tooltip'
    title="默认的 Tooltip">
    默认的 Tooltip
</button>
<button type="button" class="btn btn-default" data-toggle="tooltip'
    data-placement="left" title="左侧的 Tooltip">
    左侧的 Tooltip
</button>
<button type="button" class="btn btn-default" data-toggle="tooltip'
    data-placement="top" title="顶部的 Tooltip">
    顶部的 Tooltip
</button>
<button type="button" class="btn btn-default" data-toggle="tooltip'
    data-placement="bottom" title="底部的 Tooltip">
    底部的 Tooltip
</button>
<button type="button" class="btn btn-default" data-toggle="tooltip'
    data-placement="right" title="右侧的 Tooltip">
    右侧的 Tooltip
</button>
<script>
    $(function () { $("[data-toggle='tooltip']").tooltip(); });
```

```
</script>

</body>
</html>
```

结果如下所示：

工具提示（Tooltip）插件 - 锚

这是一个 默认的 Tooltip . 这是一个 左侧的 Tooltip . 这是一个 顶部的 Tooltip . 这是一个 底部的 Tooltip . 这是一个 右侧的 Tooltip

工具提示（Tooltip）插件 - 按钮

默认的 Tooltip | 左侧的 Tooltip | 顶部的 Tooltip | 底部的 Tooltip | 右侧的 Tooltip

选项

有一些选项是通过 Bootstrap 数据 API（Bootstrap Data API）添加或通过 JavaScript 调用的。下表列出了这些选项：

选项名称	类型/默认值	Data 属性名称	描述
animation	boolean 默认值：*true*	data-animation	工具提示使用 CSS 渐变滤镜效果。
html	boolean 默认值：*false*	data-html	向工具提示插入 HTML。如果为 fals jQuery 的 text 方法将被用于向 dom 容。如果您担心 XSS 攻击，请使用
placement	string|function 默认值：*top*	data-placement	规定如何定位工具提示（即 top|bottom|left|right|auto）。当指定时，会动态调整工具提示。例如，如 placement 是 "auto left"，工具提示 可能显示在左边，在情况不允许的情 才会显示在右边。
selector	string 默认值：*false*	data-selector	如果提供了一个选择器，工具提示对 委派到指定的目标。
title	string | function 默认值：*''*	data-title	如果未指定 *title* 属性，则 title 选项 的 title 值。
trigger	string 默认值：*'hover focus'*	data-trigger	定义如何触发工具提示： **click| hov focus | manual**。您可以传递多个触 每个触发器之间用空格分隔。
content	string | function 默认值：*''*	data-content	如果未指定 *data-content* 属性，则使 认的 content 值。
delay	number | object 默认值：*0*	data-delay	延迟显示和隐藏工具提示的毫秒数 - manual 手动触发类型不适用。如果 是一个数字，那么延迟将会应用于显 藏。如果提供的是对象，结构如下所 示： `delay:{ show: 500, hide:`
container	string | false 默认值：*false*	data-container	向指定元素追加工具提示。实例： container: 'body'

方法

下面是一些工具提示（Tooltip）插件中有用的方法：

方法	描述	实例
Options: .tooltip(options)	向元素集合附加提示工具句柄。	`$().tooltip(options)`
Toggle: .tooltip('toggle')	切换显示/隐藏元素的提示工具。	`$('#element').tooltip('toggle')`
Show: .tooltip('show')	显示元素的提示工具。	`$('#element').tooltip('show')`
Hide: .tooltip('hide')	隐藏元素的提示工具。	`$('#element').tooltip('hide')`
Destroy: .tooltip('destroy')	隐藏并销毁元素的提示工具。	`$('#element').tooltip('destroy')`

实例

下面的实例演示了工具提示（Tooltip）插件方法的用法。

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 工具提示（Tooltip）插件方法</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div style="padding: 100px 100px 10px;">
   这是一个 <a href="#" class="tooltip-show" data-toggle="tooltip"
      title="show">Tooltip 方法 show
   </a>.

   这是一个 <a href="#" class="tooltip-hide" data-toggle="tooltip"
      data-placement="left" title="hide">Tooltip 方法 hide
   </a>.

   这是一个 <a href="#" class="tooltip-destroy" data-toggle="tooltip
      data-placement="top" title="destroy">Tooltip 方法 destroy
   </a>.

   这是一个 <a href="#" class="tooltip-toggle" data-toggle="tooltip"
      data-placement="bottom" title="toggle">Tooltip 方法 toggle
   </a>.
   <br><br><br><br><br><br>
   <p class="tooltip-options" >
      这是一个 <a href="#" data-toggle="tooltip" title="<h2>'am Head
         </h2>">Tooltip 方法 options
      </a>.
   </p>

   <script>
      $(function () { $('.tooltip-show').tooltip('show');});
      $(function () { $('.tooltip-hide').tooltip('hide');});
      $(function () { $('.tooltip-destroy').tooltip('destroy');});
      $(function () { $('.tooltip-toggle').tooltip('toggle');});
      $(function () { $(".tooltip-options a").tooltip({html : true
      });
   </script>
<div>

</body>
</html>
```

结果如下所示：



事件

下表列出了工具提示（Tooltip）插件中要用到的事件。这些事件可在函数中当钩子使用。

事件	描述	实例
show.bs.tooltip	当调用 show 实例方法时立即触发该事件。	`$('#myTooltip').on('show.bs.tooltip', fun`
shown.bs.tooltip	当工具提示对用户可见时触发该事件（将等待 CSS 过渡效果完成）。	`$('#myTooltip').on('shown.bs.tooltip', fu`
hide.bs.tooltip	当调用 hide 实例方法时立即触发该事件。	`$('#myTooltip').on('hide.bs.tooltip', fun`
hidden.bs.tooltip	当工具提示对用户隐藏时触发该事件（将等待 CSS 过渡效果完成）。	`$('#myTooltip').on('hidden.bs.tooltip', f`

实例

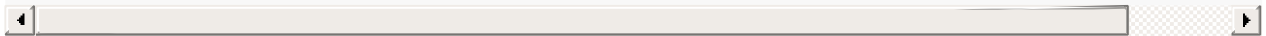
下面的实例演示了工具提示（Tooltip）插件事件的用法。

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 工具提示（Tooltip）插件事件</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

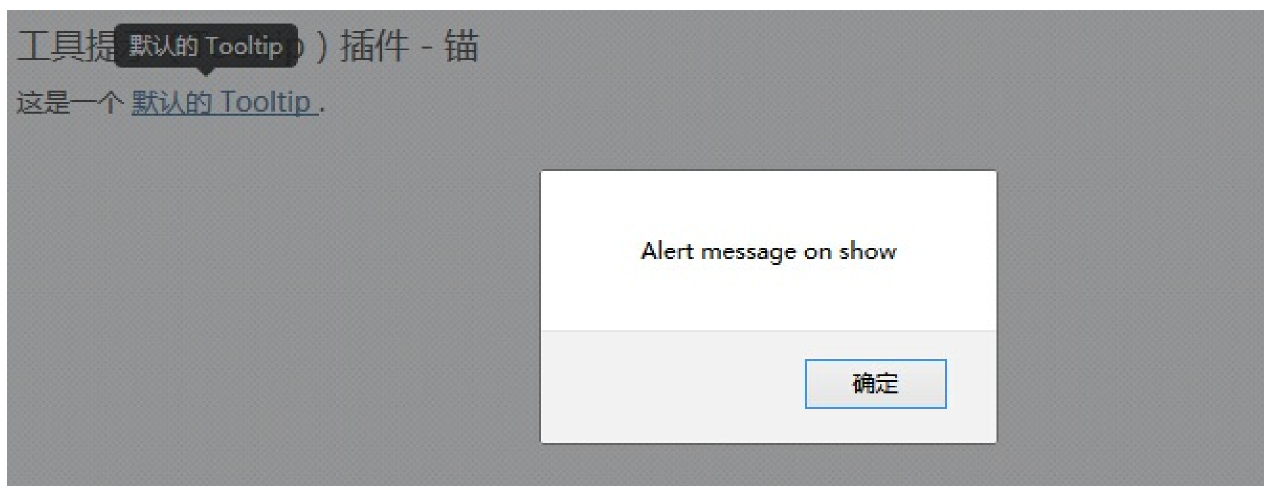
<h4>工具提示（Tooltip）插件 - 锚</h4>
这是一个 <a href="#" class="tooltip-show" data-toggle="tooltip"
   title="默认的 Tooltip">默认的 Tooltip
</a>.

<script>
$(function () { $('.tooltip-show').tooltip('show');});
$(function () { $('.tooltip-show').on('show.bs.tooltip', function (
   alert("Alert message on show");
})});
</script>

</body>
</html>
```



结果如下所示：



Bootstrap 弹出框（Popover）插件

弹出框（Popover）与工具提示（Tooltip）类似，提供了一个扩展的视图。如需激活弹出框，用户只需把鼠标悬停在元素上即可。弹出框的内容完全可使用 Bootstrap 数据 API（Bootstrap Data API）来填充。该方法依赖于工具提示（tooltip）。

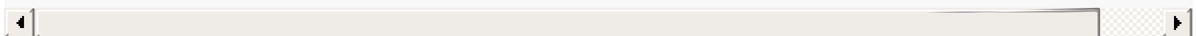
如果您想要单独引用该插件的功能，那么您需要引用 **popover.js**，它依赖于 工具提示（Tooltip）插件。或者，正如 Bootstrap 插件概览 一章中所提到，您可以引用 *bootstrap.js* 或压缩版的 *bootstrap.min.js*。

用法

弹出框（Popover）插件根据需求生成内容和标记，默认情况下是把弹出框（popover）放在它们的触发元素后面。您可以有以下两种方式添加弹出框（popover）：

- 通过 **data** 属性：如需添加一个弹出框（popover），只需向一个锚/按钮标签添加 **data-toggle="popover"** 即可。锚的 title 即为弹出框（popover）的文本。默认情况下，插件把弹出框（popover）设置在顶部。

```
&lt;a href="#" data-toggle="popover" title="Example popover"&gt
   请悬停在我的上面
&lt;/a&gt;
```



- 通过 **JavaScript**：通过 JavaScript 启用弹出框（popover）：

```
$('#identifier').popover(options)
```

弹出框（Popover）插件不像之前所讨论的下拉菜单及其他插件那样，它不是纯 CSS 插件。如需使用该插件，您必须使用 jquery 激活它（读取 javascript）。使用下面的脚本来启用页面中的所有的弹出框（popover）：

```
$(function () { $("[data-toggle='popover']").popover(); });
```

实例

下面的实例演示了通过 data 属性使用弹出框（Popover）插件的用法。

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 弹出框（Popover）插件</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="container" style="padding: 100px 50px 10px;" >
   <button type="button" class="btn btn-default" title="Popover ti
      data-container="body" data-toggle="popover" data-placement="
      data-content="左侧的 Popover 中的一些内容">
      左侧的 Popover
   </button>
   <button type="button" class="btn btn-primary" title="Popover ti
      data-container="body" data-toggle="popover" data-placement="
      data-content="顶部的 Popover 中的一些内容">
      顶部的 Popover
   </button>
   <button type="button" class="btn btn-success" title="Popover ti
      data-container="body" data-toggle="popover" data-placement="
      data-content="底部的 Popover 中的一些内容">
      底部的 Popover
   </button>
   <button type="button" class="btn btn-warning" title="Popover ti
      data-container="body" data-toggle="popover" data-placement="
      data-content="右侧的 Popover 中的一些内容">
      右侧的 Popover
   </button>
   </div>

   <script>$(function ()
      { $("[data-toggle='popover']").popover();
      });
   </script>
</div>

</body>
</html>
```

结果如下所示：



选项

有一些选项是通过 Bootstrap 数据 API（Bootstrap Data API）添加或通过 JavaScript 调用的。下表列出了这些选项：

选项名称	类型/默认值	Data 属性名称	描述
animation	boolean 默认值：*true*	data-animation	向弹出框应用 CSS 褪色过渡效果。
html	boolean 默认值：*false*	data-html	向弹出框插入 HTML。如果为 false, jQuery 的 text 方法将被用于向 dom 容。如果您担心 XSS 攻击，请使用
placement	string|function 默认值：*top*	data-placement	规定如何定位弹出框（即 top|bottom|left|right|auto）。当指定时，会动态调整弹出框。例如，如果 placement 是 "auto left"，弹出框将 能显示在左边，在情况不允许的情况 会显示在右边。
selector	string 默认值：*false*	data-selector	如果提供了一个选择器，弹出框对象 派到指定的目标。
title	string | function 默认值：*''*	data-title	如果未指定 *title* 属性，则 title 选项是 的 title 值。
trigger	string 默认值：*'hover focus'*	data-trigger	定义如何触发弹出框：**click| hover focus | manual**。您可以传递多个触 每个触发器之间用空格分隔。
delay	number | object 默认值：*0*	data-delay	延迟显示和隐藏弹出框的毫秒数 - 对 manual 手动触发类型不适用。如果 是一个数字，那么延迟将会应用于显 藏。如果提供的是对象，结构如下所 示：`delay:{ show: 500, hide:`
container	string | false 默认值：*false*	data-container	向指定元素追加弹出框。实例：cor 'body'

方法

下面是一些弹出框（Popover）插件中有用的方法：

方法	描述	实例
Options: .popover(options)	向元素集合附加弹出框句柄。	`$().popover(options)`
Toggle: .popover('toggle')	切换显示/隐藏元素的弹出框。	`$('#element').popover('toggle')`
Show: .popover('show')	显示元素的弹出框。	`$('#element').popover('show')`
Hide: .popover('hide')	隐藏元素的弹出框。	`$('#element').popover('hide')`
Destroy: .popover('destroy')	隐藏并销毁元素的弹出框。	`$('#element').popover('destroy')`

实例

下面的实例演示了弹出框（Popover）插件的方法：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 弹出框（Popover）插件方法</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="container" style="padding: 100px 50px 10px;" >
   <button type="button" class="btn btn-default popover-show"
      title="Popover title" data-container="body"
      data-toggle="popover" data-placement="left"
      data-content="左侧的 Popover 中的一些内容 —— show 方法">
      左侧的 Popover
   </button>
   <button type="button" class="btn btn-primary popover-hide"
      title="Popover title" data-container="body"
      data-toggle="popover" data-placement="top"
      data-content="顶部的 Popover 中的一些内容 —— hide 方法">
      顶部的 Popover
```

```
    </button>
    <button type="button" class="btn btn-success popover-destroy"
       title="Popover title" data-container="body"
       data-toggle="popover" data-placement="bottom"
       data-content="底部的 Popover 中的一些内容 —— destroy 方法">
       底部的 Popover
    </button>
    <button type="button" class="btn btn-warning popover-toggle"
       title="Popover title" data-container="body"
       data-toggle="popover" data-placement="right"
       data-content="右侧的 Popover 中的一些内容 —— toggle 方法">
       右侧的 Popover
    </button><br><br><br><br><br><br>
    <p class="popover-options">
       <a href="#" type="button" class="btn btn-warning" title="<h2>
          data-container="body" data-toggle="popover" data-content="
          <h4>Popover 中的一些内容 —— options 方法</h4>">
          Popover
       </a>
    </p>
    <script>
       $(function () { $('.popover-show').popover('show');});
       $(function () { $('.popover-hide').popover('hide');});
       $(function () { $('.popover-destroy').popover('destroy');});
       $(function () { $('.popover-toggle').popover('toggle');});
      $(function () { $(".popover-options a").popover({html : true ]
    </script>
 </div>

 </body>
 </html>
```

结果如下所示：



事件

下表列出了弹出框（Popover）插件中要用到的事件。这些事件可在函数中当钩子使用。

事件	描述	实例
show.bs.popover	当调用 show 实例方法时立即触发该事件。	`$('#mypopover').on('show.bs.popover', f`
shown.bs.popover	当弹出框对用户可见时触发该事件（将等待 CSS 过渡效果完成）。	`$('#mypopover').on('shown.bs.popover',`
hide.bs.popover	当调用 hide 实例方法时立即触发该事件。	`$('#mypopover').on('hide.bs.popover', f`
hidden.bs.popover	当工具提示对用户隐藏时触发该事件（将等待 CSS 过渡效果完成）。	`$('#mypopover').on('hidden.bs.popover',`

实例

下面的实例演示了弹出框（Popover）插件的事件：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 弹出框（Popover）插件事件</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="container" style="padding: 100px 50px 10px;" >
   <button type="button" class="btn btn-primary popover-show"
      title="Popover title" data-container="body"
      data-toggle="popover"
      data-content="左侧的 Popover 中的一些内容 —— show 方法">
      左侧的 Popover
   </button>

   </div>
   <script>
      $(function () { $('.popover-show').popover('show');});
      $(function () { $('.popover-show').on('shown.bs.popover', fur
      alert("当显示时警告消息");
   })});
   </script>
</div>

</body>
</html>
```

结果如下所示：



Bootstrap 警告框（Alert）插件

警告框（Alert）消息大多是用来想终端用户显示诸如警告或确认消息的信息。使用警告框（Alert）插件，您可以向所有的警告框消息添加可取消（dismiss）功能。

> 如果您想要单独引用该插件的功能，那么您需要引用 **alert.js**。或者，正如 Bootstrap 插件概览 一章中所提到，您可以引用 *bootstrap.js* 或压缩版的 *bootstrap.min.js*。

用法

您可以有以下两种方式启用警告框的可取消（dismissal）功能：

- 通过 **data** 属性：通过数据 API（Data API）添加可取消功能，只需要向关闭按钮添加 **data-dismiss="alert"**，就会自动为警告框添加关闭功能。

```
&lt;a class="close" data-dismiss="alert" href="#" aria-hidden="
   &times;
&lt;/a&gt;
```

- 通过 **JavaScript**：通过 JavaScript 添加可取消功能：

```
$(".alert").alert()
```

实例

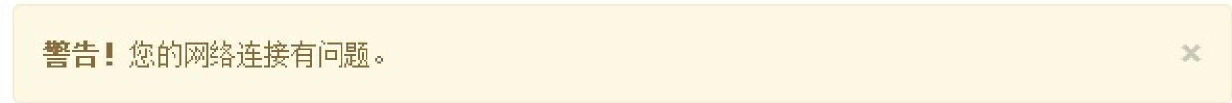
下面的实例演示了通过 data 属性使用警告框（Alert）插件的用法。

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 警告框（Alert）插件</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="alert alert-warning">
   <a href="#" class="close" data-dismiss="alert">
      &times;
   </a>
   <strong>警告！</strong>您的网络连接有问题。
</div>

</body>
</html>
```

结果如下所示：



选项

没有选项。

方法

下面是一些警告框（Alert）插件中有用的方法：

方法	描述	实例
.alert()	该方法让所有的警告框都带有关闭功能。	`$('#identifier').alert();`
Close Method .alert('close')	关闭所有的警告框。	`$('#identifier').alert('close');`

> 如需在关闭时启用动画效果，请确保添加了 **.fade** 和 **.in** class。

实例

下面的实例演示了 **.alert()** 方法的用法：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 警告框（Alert）插件 alert() 方法</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<h3>警告框（Alert）插件 alert() 方法</h3>
<div id="myAlert" class="alert alert-success">
   <a href="#" class="close" data-dismiss="alert">&times;</a>
   <strong>成功！</strong>结果是成功的。
</div>
<div id="myAlert" class="alert alert-warning">
   <a href="#" class="close" data-dismiss="alert">&times;</a>
   <strong>警告！</strong>您的网络连接有问题。
</div>

<script type="text/javascript">
$(function(){
   $(".close").click(function(){
      $("#myAlert").alert();
   });
});
</script>

</body>
</html>
```

下面的实例演示了 **.alert('close')** 方法的用法：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 警告框（Alert）插件 alert('close') 方法</ti
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<h3>警告框（Alert）插件 alert('close') 方法</h3>
<div id="myAlert" class="alert alert-success">
   <a href="#" class="close" data-dismiss="alert">&times;</a>
   <strong>成功！</strong>结果是成功的。
</div>
<div id="myAlert" class="alert alert-warning">
   <a href="#" class="close" data-dismiss="alert">&times;</a>
   <strong>警告！</strong>您的网络连接有问题。
</div>

<script type="text/javascript">
$(function(){
   $(".close").click(function(){
      $("#myAlert").alert('close');
   });
});

</script>

</body>
</html>
```

您可以看到所有的警告框都应用了关闭功能，即关闭任意的警告框，其他剩余的警告框也会被关闭。

事件

下表列出了警告框（Alert）插件中要用到的事件。这些事件可在函数中当钩子使用。

事件	描述	实例
close.bs.alert	当调用 *close* 实例方法时立即触发该事件。	`$('#myalert').bind('close.bs.alert', functi`
closed.bs.alert	当警告框被关闭时触发该事件（将等待 CSS 过渡效果完成）。	`$('#myalert').bind('closed.bs.alert', funct`

实例

下面的实例演示了警告框（Alert）插件的事件：

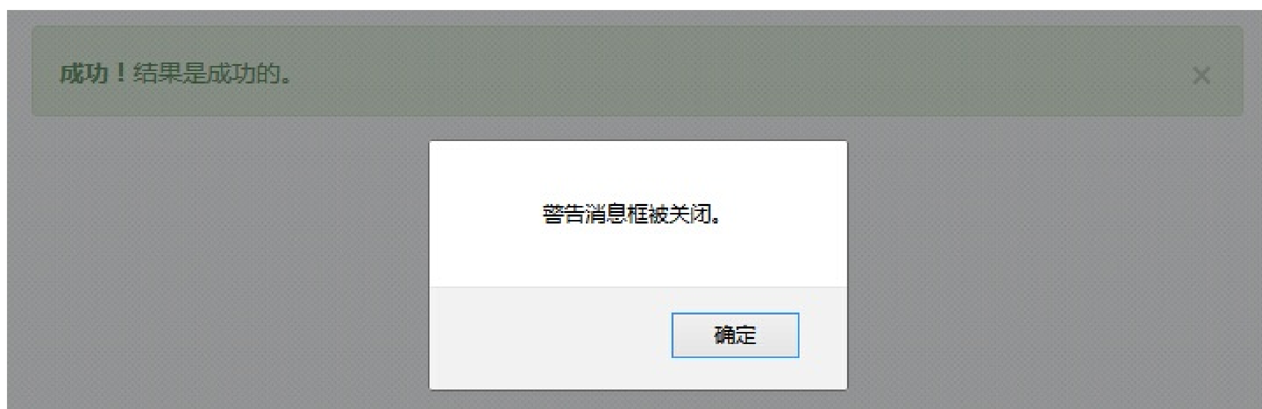
```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 警告框（Alert）插件事件</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div id="myAlert" class="alert alert-success">
   <a href="#" class="close" data-dismiss="alert">&times;</a>
   <strong>成功！</strong>结果是成功的。
</div>

<script type="text/javascript">
$(function(){
   $("#myAlert").bind('closed.bs.alert', function () {
      alert("警告消息框被关闭。");
   });
});
</script>

</body>
</html>
```

结果如下所示：



Bootstrap 按钮（Button）插件

按钮（Button）在 Bootstrap 按钮 一章中介绍过。通过按钮（Button）插件，您可以添加进一些交互，比如控制按钮状态，或者为其他组件（如工具栏）创建按钮组。

> 如果您想要单独引用该插件的功能，那么您需要引用 **button.js**。或者，正如 Bootstrap 插件概览 一章中所提到，您可以引用 *bootstrap.js* 或压缩版的 *bootstrap.min.js*。

加载状态

如需向按钮添加加载状态，只需要简单地向 button 元素添加 **data-loading-text="Loading..."** 作为其属性即可，如下面实例所示：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 按钮（Button）插件加载状态</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<button id="fat-btn" class="btn btn-primary" data-loading-text="Loa
   type="button"> 加载状态
</button>
<script>
   $(function() {
      $(".btn").click(function(){
         $(this).button('loading').delay(1000).queue(function() {
           // $(this).button('reset');
         });
      });
   });
</script>

</body>
</html>
```

结果如下所示：

Loading...

单个切换

如需激活单个按钮的切换（即改变按钮的正常状态为按压状态，反之亦然），只需向 button 元素添加 **data-toggle="button"** 作为其属性即可，如下面实例所示：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 按钮（Button）插件单个切换</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<button type="button" class="btn btn-primary"
   data-toggle="button"> 单个切换
</button>

</body>
</html>
```

结果如下所示：

单个切换

复选框（Checkbox）

您可以创建复选框组，并通过向 **btn-group** 添加 data 属性 **data-toggle="buttons"** 来添加复选框组的切换。

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 按钮（Button）插件复选框</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="btn-group" data-toggle="buttons">
   <label class="btn btn-primary">
      <input type="checkbox"> 选项 1
   </label>
   <label class="btn btn-primary">
      <input type="checkbox"> 选项 2
   </label>
   <label class="btn btn-primary">
      <input type="checkbox"> 选项 3
   </label>
</div>

</body>
</html>
```

结果如下所示：



单选按钮（Radio）

类似地，您可以创建单选按钮组，并通过向 **btn-group** 添加 data 属性 **data-toggle="buttons"** 来添加单选按钮组的切换。

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 按钮（Button）插件单选按钮</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="btn-group" data-toggle="buttons">
   <label class="btn btn-primary">
      <input type="radio" name="options" id="option1"> 选项 1
   </label>
   <label class="btn btn-primary">
      <input type="radio" name="options" id="option2"> 选项 2
   </label>
   <label class="btn btn-primary">
      <input type="radio" name="options" id="option3"> 选项 3
   </label>
</div>

</body>
</html>
```

结果如下所示：



用法

您可以 通过 **JavaScript** 启用按钮（Button）插件，如下所示：

```
$('.btn').button()
```

选项

没有选项。

方法

下面是一些按钮（Button）插件中有用的方法：

方法	描述	实例
button('toggle')	切换按压状态。赋予按钮被激活的外观。您可以使用 **data-toggle** 属性启用按钮的自动切换。	`$().button('toggle')`
.button('loading')	当加载时，按钮是禁用的，且文本变为 button 元素的 **data-loading-text** 属性的值。	`$().button('loading')`
.button('reset')	重置按钮状态，文本内容恢复为最初的内容。当您想要把按钮返回为原始的状态时，该方法非常有用。	`$().button('reset')`
.button(string)	该方法中的字符串是指由用户声明的任何字符串。使用该方法，重置按钮状态，并添加新的内容。	`$().button(string)`

实例

下面的实例演示了上面方法的用法：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 按钮（Button）插件方法</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<h2>点击每个按钮查看方法效果</h2>
<h4>演示 .button('toggle') 方法</h4>
<div id="myButtons1" class="bs-example">
   <button type="button" class="btn btn-primary">原始</button>
</div>

<h4>演示 .button('loading') 方法</h4>
<div id="myButtons2" class="bs-example">
   <button type="button" class="btn btn-primary"
      data-loading-text="Loading...">原始
   </button>
</div>

<h4>演示 .button('reset') 方法</h4>
```

```
<div id="myButtons3" class="bs-example">
   <button type="button" class="btn btn-primary"
      data-loading-text="Loading...">原始
   </button>
</div>

<h4>演示 .button(string) 方法</h4>
<button type="button" class="btn btn-primary" id="myButton4"
   data-complete-text="Loading finished">请点击我
</button>
<script type="text/javascript">
   $(function () {
      $("#myButtons1 .btn").click(function(){
         $(this).button('toggle');
      });
   });
   $(function() {
      $("#myButtons2 .btn").click(function(){
         $(this).button('loading').delay(1000).queue(function() {
         });
      });
   });
   $(function() {
      $("#myButtons3 .btn").click(function(){
         $(this).button('loading').delay(1000).queue(function() {
            $(this).button('reset');
         });
      });
   });
   $(function() {
      $("#myButton4").click(function(){
         $(this).button('loading').delay(1000).queue(function() {
            $(this).button('complete');
         });
      });
   });
</script>

</body>
</html>
```

结果如下所示：

点击每个按钮查看方法效果

演示 .button('toggle') 方法

原始

演示 .button('loading') 方法

原始

演示 .button('reset') 方法

原始

演示 .button(string) 方法

请点击我

Bootstrap 折叠（Collapse）插件

折叠（Collapse）插件可以很容易地让页面区域折叠起来。无论您用它来创建折叠导航还是内容面板，它都允许很多内容选项。

> 如果您想要单独引用该插件的功能，那么您需要引用 **collapse.js**。同时，也需要在您的 Bootstrap 版本中引用 Transition（过渡）插件。或者，正如 Bootstrap 插件概览 一章中所提到，您可以引用 *bootstrap.js* 或压缩版的 *bootstrap.min.js*。

您可以使用折叠（Collapse）插件：

- 创建可折叠的分组或折叠面板（***accordion***），如下所示：

```
&lt;!DOCTYPE html&gt;
&lt;html&gt;
&lt;head&gt;
   &lt;title&gt;Bootstrap 实例 - 折叠面板&lt;/title&gt;
   &lt;link href="/bootstrap/css/bootstrap.min.css" rel="styles
   &lt;script src="/scripts/jquery.min.js"&gt;&lt;/script&gt;
   &lt;script src="/bootstrap/js/bootstrap.min.js"&gt;&lt;/scri
&lt;/head&gt;
&lt;body&gt;

&lt;div class="panel-group" id="accordion"&gt;
  &lt;div class="panel panel-default"&gt;
    &lt;div class="panel-heading"&gt;
      &lt;h4 class="panel-title"&gt;
        &lt;a data-toggle="collapse" data-parent="#accordion"
          href="#collapseOne"&gt;
          点击我进行展开，再次点击我进行折叠。第 1 部分
        &lt;/a&gt;
      &lt;/h4&gt;
    &lt;/div&gt;
    &lt;div id="collapseOne" class="panel-collapse collapse in"
      &lt;div class="panel-body"&gt;
        Nihil anim keffiyeh helvetica, craft beer labore wes ar
        cred nesciunt sapiente ea proident. Ad vegan excepteur
        vice lomo.
      &lt;/div&gt;
    &lt;/div&gt;
  &lt;/div&gt;
  &lt;div class="panel panel-default"&gt;
    &lt;div class="panel-heading"&gt;
      &lt;h4 class="panel-title"&gt;
        &lt;a data-toggle="collapse" data-parent="#accordion"
          href="#collapseTwo"&gt;
          点击我进行展开，再次点击我进行折叠。第 2 部分
        &lt;/a&gt;
```

```
        &lt;/h4&gt;
      &lt;/div&gt;
      &lt;div id="collapseTwo" class="panel-collapse collapse"&gt
        &lt;div class="panel-body"&gt;
          Nihil anim keffiyeh helvetica, craft beer labore wes ar
          cred nesciunt sapiente ea proident. Ad vegan excepteur
          vice lomo.
        &lt;/div&gt;
      &lt;/div&gt;
    &lt;/div&gt;
    &lt;div class="panel panel-default"&gt;
      &lt;div class="panel-heading"&gt;
        &lt;h4 class="panel-title"&gt;
          &lt;a data-toggle="collapse" data-parent="#accordion"
            href="#collapseThree"&gt;
            点击我进行展开，再次点击我进行折叠。第 3 部分
          &lt;/a&gt;
        &lt;/h4&gt;
      &lt;/div&gt;
      &lt;div id="collapseThree" class="panel-collapse collapse"&
        &lt;div class="panel-body"&gt;
          Nihil anim keffiyeh helvetica, craft beer labore wes ar
          cred nesciunt sapiente ea proident. Ad vegan excepteur
          vice lomo.
        &lt;/div&gt;
      &lt;/div&gt;
    &lt;/div&gt;
  &lt;/div&gt;

  &lt;/body&gt;
  &lt;/html&gt;
```

结果如下所示：

点击我进行展开，再次点击我进行折叠。第 1 部分

点击我进行展开，再次点击我进行折叠。第 2 部分

Nihil anim keffiyeh helvetica, craft beer labore wes anderson cred nesciunt sapiente ea proident. Ad vegan excepteur butcher vice lomo.

点击我进行展开，再次点击我进行折叠。第 3 部分

1. **data-toggle="collapse"** 添加到您想要展开或折叠的组件的链接上。
2. **href** 或 **data-target** 属性添加到父组件，它的值是子组件的 *id*。
3. **data-parent** 属性把折叠面板（accordion）的 id 添加到要展开或折叠的组件的链接上。

- 创建不带 ***accordion*** 标记的简单的可折叠组件（***collapsible***），如下所示：

```
&lt;!DOCTYPE html&gt;
&lt;html&gt;
&lt;head&gt;
   &lt;title&gt;Bootstrap 实例 - 简单的可折叠组件&lt;/title&gt;
   &lt;link href="/bootstrap/css/bootstrap.min.css" rel="styles
   &lt;script src="/scripts/jquery.min.js"&gt;&lt;/script&gt;
   &lt;script src="/bootstrap/js/bootstrap.min.js"&gt;&lt;/scri
&lt;/head&gt;
&lt;body&gt;

&lt;button type="button" class="btn btn-primary" data-toggle="c
   data-target="#demo"&gt;
   简单的可折叠组件
&lt;/button&gt;

&lt;div id="demo" class="collapse in"&gt;
  Nihil anim keffiyeh helvetica, craft beer labore wes anderson
  cred nesciunt sapiente ea proident. Ad vegan excepteur butche
  vice lomo.
&lt;/div&gt;

&lt;/body&gt;
&lt;/html&gt;
```

结果如下所示：



正如您在实例中看到的，我们创建了一个可折叠的组件，与折叠面板（accordion）不同，我们没有添加属性 **data-parent**。

用法

下表列出了折叠（Collapse）插件用于处理繁重的伸缩的 class：

Class	描述
.collapse	隐藏内容。
.collapse.in	显示内容。
.collapsing	当过渡效果开始时被添加，当过渡效果完成时被移除。

您可以通过以下两种方式使用折叠（Collapse）插件：

- 通过 **data 属性**：向元素添加 **data-toggle="collapse"** 和 **data-target**，自动分配可折叠元素的控制。**data-target** 属性接受一个 CSS 选择器，并会对其应用折叠效果。请确保向可折叠元素添加 class **.collapse**。如果您希望它默认情况下是打开的，请添加额外的 class **.in**。

  为了向可折叠控件添加类似折叠面板的分组管理，请添加 data 属性 **data-parent="#selector"**。

- 通过 **JavaScript**：可通过 JavaScript 激活 collapse 方法，如下所示：

```
$('.collapse').collapse()
```

选项

有一些选项是通过 data 属性或 JavaScript 来传递的。下表列出了这些选项：

选项名称	类型/默认值	Data 属性名称	描述
parent	selector 默认值：*false*	data-parent	如果提供了一个选择器，当可折叠项目显示时，指定父元素下的所有可折叠的元素将被关闭。这与创痛的折叠面板（accordion）的行为类似 - 这依赖于 accordion-group 类。
toggle	boolean 默认值：*true*	data-toggle	切换调用可折叠元素。

方法

下面是一些折叠（Collapse）插件中有用的方法：

方法	描述	实例
Options: .collapse(options)	激活内容为可折叠元素。接受一个可选的 options 对象。	`$('#identifier').collapse({ toggle: fals`
Toggle: .collapse('toggle')	切换显示/隐藏可折叠元素。	`$('#identifier').collapse('toggle')`
Show: .collapse('show')	显示可折叠元素。	`$('#identifier').collapse('show')`
Hide: .collapse('hide')	隐藏可折叠元素。	`$('#identifier').collapse('hide')`

实例

下面的实例演示了方法的用法：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 折叠（Collapse）插件方法</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="panel-group" id="accordion">
   <div class="panel panel-default">
      <div class="panel-heading">
         <h4 class="panel-title">
            <a data-toggle="collapse" data-parent="#accordion"
               href="#collapseOne">
               点击我进行展开，再次点击我进行折叠。第 1 部分--hide 方法
            </a>
         </h4>
      </div>
      <div id="collapseOne" class="panel-collapse collapse in">
```

```
            <div class="panel-body">
               Nihil anim keffiyeh helvetica, craft beer labore wes ar
               nesciunt sapiente ea proident. Ad vegan excepteur butch
               lomo.
            </div>
         </div>
      </div>
      <div class="panel panel-success">
         <div class="panel-heading">
            <h4 class="panel-title">
               <a data-toggle="collapse" data-parent="#accordion"
                  href="#collapseTwo">
                  点击我进行展开，再次点击我进行折叠。第 2 部分--show 方法
               </a>
            </h4>
         </div>
         <div id="collapseTwo" class="panel-collapse collapse">
            <div class="panel-body">
               Nihil anim keffiyeh helvetica, craft beer labore wes ar
               nesciunt sapiente ea proident. Ad vegan excepteur butch
               lomo.
            </div>
         </div>
      </div>
      <div class="panel panel-info">
         <div class="panel-heading">
            <h4 class="panel-title">
               <a data-toggle="collapse" data-parent="#accordion"
                  href="#collapseThree">
                  点击我进行展开，再次点击我进行折叠。第 3 部分--toggle 方法
               </a>
            </h4>
         </div>
         <div id="collapseThree" class="panel-collapse collapse">
            <div class="panel-body">
               Nihil anim keffiyeh helvetica, craft beer labore wes ar
               nesciunt sapiente ea proident. Ad vegan excepteur butch
               lomo.
            </div>
         </div>
      </div>
      <div class="panel panel-warning">
         <div class="panel-heading">
            <h4 class="panel-title">
               <a data-toggle="collapse" data-parent="#accordion"
                  href="#collapseFour">
                  点击我进行展开，再次点击我进行折叠。第 4 部分--options 方法
               </a>
            </h4>
         </div>
         <div id="collapseFour" class="panel-collapse collapse">
            <div class="panel-body">
               Nihil anim keffiyeh helvetica, craft beer labore wes ar
```

```
              nesciunt sapiente ea proident. Ad vegan excepteur butch
              lomo.
           </div>
        </div>
     </div>
  </div>
  <script type="text/javascript">
     $(function () { $('#collapseFour').collapse({
        toggle: false
     })});
     $(function () { $('#collapseTwo').collapse('show')});
     $(function () { $('#collapseThree').collapse('toggle')});
     $(function () { $('#collapseOne').collapse('hide')});
  </script>

  </body>
  </html>
```

结果如下所示：

点击我进行展开，再次点击我进行折叠。第 1 部分--hide 方法

点击我进行展开，再次点击我进行折叠。第 2 部分--show 方法

Nihil anim keffiyeh helvetica, craft beer labore wes anderson cred nesciunt sapiente ea proident. Ad vegan excepteur butcher vice lomo.

点击我进行展开，再次点击我进行折叠。第 3 部分--toggle 方法

Nihil anim keffiyeh helvetica, craft beer labore wes anderson cred nesciunt sapiente ea proident. Ad vegan excepteur butcher vice lomo.

点击我进行展开，再次点击我进行折叠。第 4 部分--options 方法

事件

下表列出了折叠（Collapse）插件中要用到的事件。这些事件可在函数中当钩子使用。

事件	描述	实例
show.bs.collapse	在调用 show 方法后触发该事件。	`$('#identifier').on('show.bs.collapse',`
shown.bs.collapse	当折叠元素对用户可见时触发该事件（将等待 CSS 过渡效果完成）。	`$('#identifier').on('shown.bs.collapse'`
hide.bs.collapse	当调用 hide 实例方法时立即触发该事件。	`$('#identifier').on('hide.bs.collapse',`
hidden.bs.collapse	当折叠元素对用户隐藏时触发该事件（将等待 CSS 过渡效果完成）。	`$('#identifier').on('hidden.bs.collapse`

实例

下面的实例演示了事件的用法：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 折叠（Collapse）插件事件</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="panel-group" id="accordion">
   <div class="panel panel-info">
      <div class="panel-heading">
         <h4 class="panel-title">
            <a data-toggle="collapse" data-parent="#accordion"
               href="#collapseexample">
               点击我进行展开，再次点击我进行折叠。--shown 事件
            </a>
         </h4>
      </div>
      <div id="collapseexample" class="panel-collapse collapse">
         <div class="panel-body">
            Nihil anim keffiyeh helvetica, craft beer labore wes ar
            cred nesciunt sapiente ea proident.
            Ad vegan excepteur butcher vice lomo.
         </div>
      </div>
   </div>
</div>

<script type="text/javascript">
   $(function () {
      $('#collapseexample').on('show.bs.collapse', function () {
         alert('嘿，当您展开时会提示本警告');})
   });
</script>

</body>
</html>
```

结果如下所示：

点击我进行展开，再次点击我进行折叠。--shown 事件

嘿，当您展开时会提示本警告

确定

Bootstrap 轮播（Carousel）插件

Bootstrap 轮播（Carousel）插件是一种灵活的响应式的向站点添加滑块的方式。除此之外，内容也是足够灵活的，可以是图像、内嵌框架、视频或者其他您想要放置的任何类型的内容。

> 如果您想要单独引用该插件的功能，那么您需要引用 **carousel.js**。或者，正如 Bootstrap 插件概览 一章中所提到，您可以引用 *bootstrap.js* 或压缩版的 *bootstrap.min.js*。

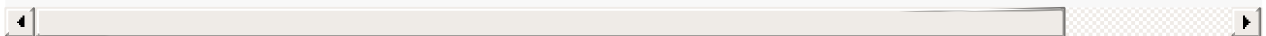
实例

下面是一个简单的幻灯片，使用 Bootstrap 轮播（Carousel）插件显示了一个循环播放元素的通用组件。为了实现轮播，您只需要添加带有该标记的代码即可。不需要使用 data 属性，只需要简单的基于 class 的开发即可。

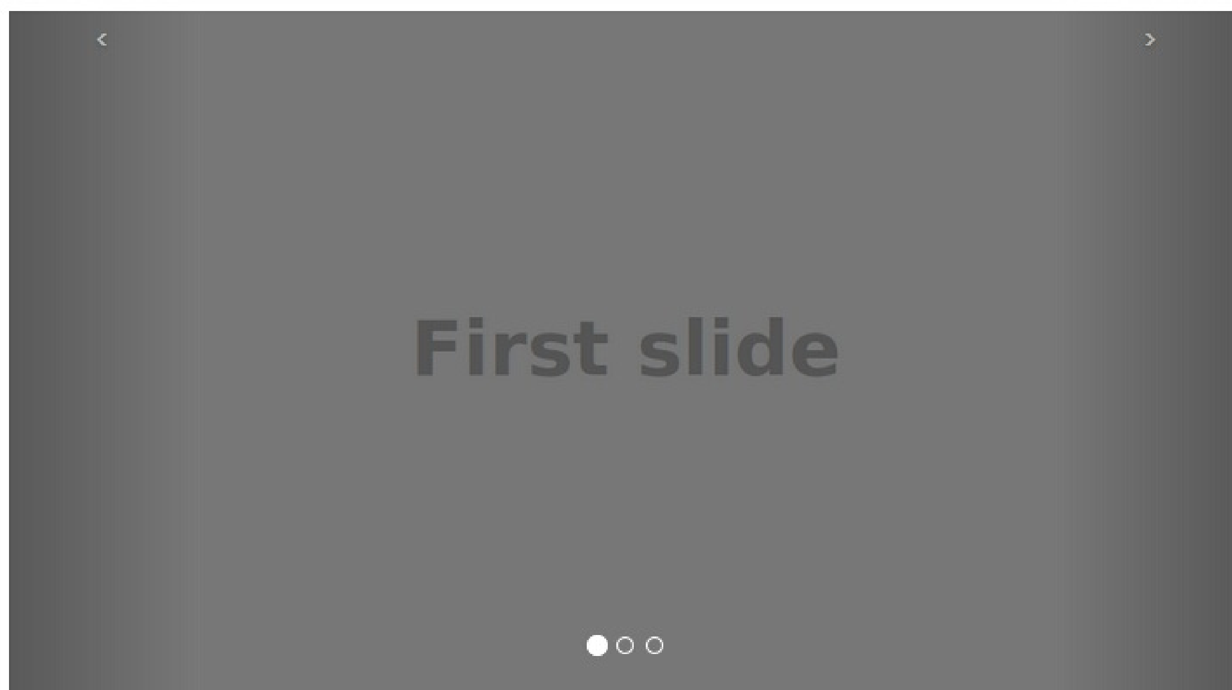
```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 简单的轮播（Carousel）插件</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div id="myCarousel" class="carousel slide">
   <!-- 轮播（Carousel）指标 -->
   <ol class="carousel-indicators">
      <li data-target="#myCarousel" data-slide-to="0" class="active
      <li data-target="#myCarousel" data-slide-to="1"></li>
      <li data-target="#myCarousel" data-slide-to="2"></li>
   </ol>
   <!-- 轮播（Carousel）项目 -->
   <div class="carousel-inner">
      <div class="item active">
         <img src="img/slide1.png" alt="First slide">
      </div>
      <div class="item">
         <img src="img/slide2.png" alt="Second slide">
      </div>
      <div class="item">
         <img src="img/slide3.png" alt="Third slide">
      </div>
   </div>
   <!-- 轮播（Carousel）导航 -->
   <a class="carousel-control left" href="#myCarousel"
      data-slide="prev">&lsaquo;</a>
   <a class="carousel-control right" href="#myCarousel"
      data-slide="next">&rsaquo;</a>
</div>

</body>
</html>
```



结果如下所示：



可选的标题

您可以通过 **.item** 内的 **.carousel-caption** 元素向幻灯片添加标题。只需要在该处放置任何可选的 HTML 即可，它会自动对齐并格式化。下面的实例演示了这点：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 轮播（Carousel）插件的标题</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div id="myCarousel" class="carousel slide">
   <!-- 轮播（Carousel）指标 -->
   <ol class="carousel-indicators">
      <li data-target="#myCarousel" data-slide-to="0" class="active
      <li data-target="#myCarousel" data-slide-to="1"></li>
      <li data-target="#myCarousel" data-slide-to="2"></li>
   </ol>
   <!-- 轮播（Carousel）项目 -->
   <div class="carousel-inner">
      <div class="item active">
         <img src="img/slide1.png" alt="First slide">
         <div class="carousel-caption">标题 1</div>
      </div>
      <div class="item">
         <img src="img/slide2.png" alt="Second slide">
         <div class="carousel-caption">标题 2</div>
      </div>
      <div class="item">
         <img src="img/slide3.png" alt="Third slide">
         <div class="carousel-caption">标题 3</div>
      </div>
   </div>
   <!-- 轮播（Carousel）导航 -->
   <a class="carousel-control left" href="#myCarousel"
      data-slide="prev">&lsaquo;</a>
   <a class="carousel-control right" href="#myCarousel"
      data-slide="next">&rsaquo;</a>
</div>

</body>
</html>
```

结果如下所示：



用法

- 通过 **data** 属性：使用 data 属性可以很容易控制轮播（Carousel）的位置。
  - 属性 **data-slide** 接受关键字 *prev* 或 *next*，用来改变幻灯片相对于当前位置的位置。
  - 使用 **data-slide-to** 来向轮播床底一个原始滑动索引，**data-slide-to="2"** 将把滑块移动到一个特定的索引，索引从 0 开始计数。
  - **data-ride="carousel"** 属性用于标记轮播在页面加载时就开始动画播放。
- 通过 **JavaScript**：轮播（Carousel）可通过 JavaScript 手动调用，如下所示：

  ```
  $('.carousel').carousel()
  ```

选项

有一些选项是通过 data 属性或 JavaScript 来传递的。下表列出了这些选项：

选项名称	类型/默认值	**Data** 属性名称	描述
interval	number 默认值：*5000*	data-interval	自动循环每个项目之间延迟的时间量。如果为 false，轮播将不会自动循环。
pause	string 默认值：*"hover"*	data-pause	鼠标进入时暂停轮播循环，鼠标离开时恢复轮播循环。
wrap	boolean 默认值：*true*	data-wrap	轮播是否连续循环。

方法

下面是一些轮播（Carousel）插件中有用的方法：

方法	描述	实例
.carousel(options)	初始化轮播为可选的 options 对象，并开始循环项目。	`$('#identifier').carousel({ interval: 2`
.carousel('cycle')	从左到右循环轮播项目。	`$('#identifier').carousel('cycle')`
.carousel('pause')	停止轮播循环项目。	`$('#identifier')..carousel('pause')`
.carousel(number)	循环轮播到某个特定的帧（从 0 开始计数，与数组类似）。	`$('#identifier').carousel(number)`
.carousel('prev')	循环轮播到上一个项目。	`$('#identifier').carousel('prev')`
.carousel('next')	循环轮播到下一个项目。	`$('#identifier').carousel('next')`

实例

下面的实例演示了方法的用法：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 轮播（Carousel）插件方法</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
```

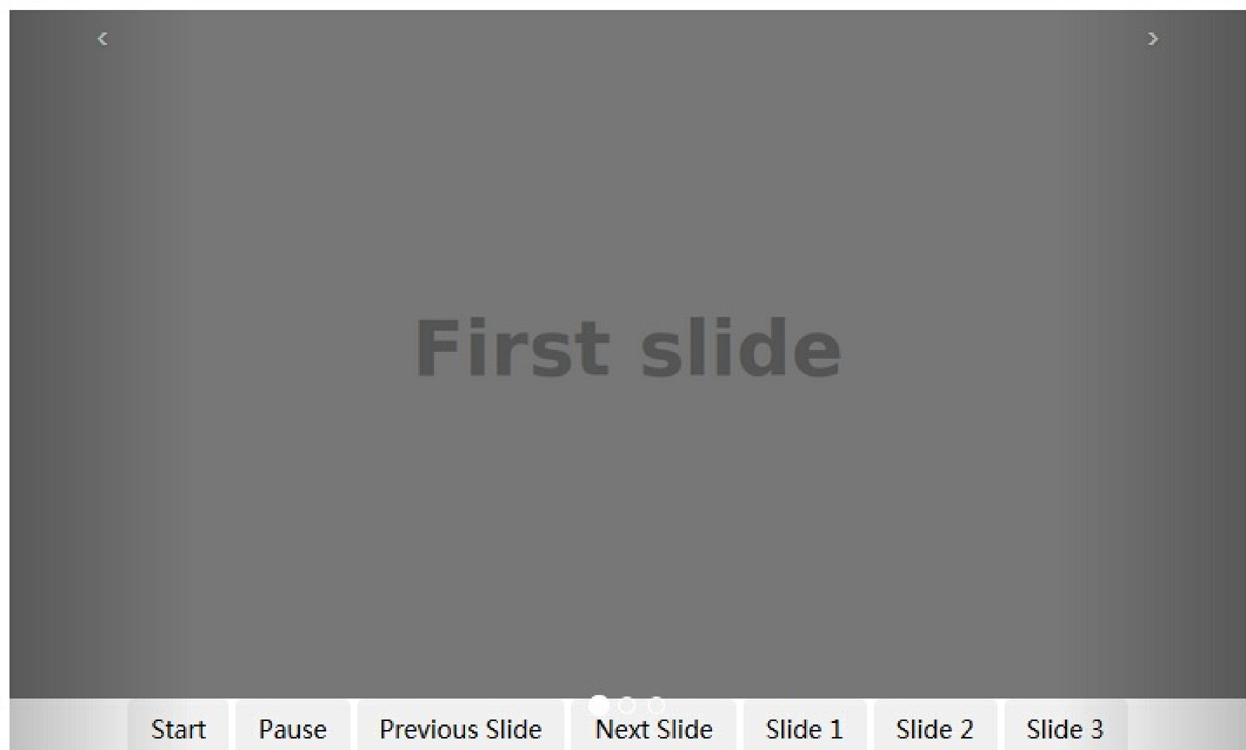
```
    <script src="/scripts/jquery.min.js"></script>
    <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div id="myCarousel" class="carousel slide">
    <!-- 轮播（Carousel）指标 -->
    <ol class="carousel-indicators">
        <li data-target="#myCarousel" data-slide-to="0"
            class="active"></li>
        <li data-target="#myCarousel" data-slide-to="1"></li>
        <li data-target="#myCarousel" data-slide-to="2"></li>
    </ol>
    <!-- 轮播（Carousel）项目 -->
    <div class="carousel-inner">
        <div class="item active">
            <img src="img/slide1.png" alt="First slide">
        </div>
        <div class="item">
            <img src="img/slide2.png" alt="Second slide">
        </div>
        <div class="item">
            <img src="img/slide3.png" alt="Third slide">
        </div>
    </div>
    <!-- 轮播（Carousel）导航 -->
    <a class="carousel-control left" href="#myCarousel"
        data-slide="prev">&lsaquo;</a>
    <a class="carousel-control right" href="#myCarousel"
        data-slide="next">&rsaquo;</a>
    <!-- 控制按钮 -->
    <div style="text-align:center;">
        <input type="button" class="btn start-slide" value="Start">
        <input type="button" class="btn pause-slide" value="Pause">
        <input type="button" class="btn prev-slide" value="Previous S
        <input type="button" class="btn next-slide" value="Next Slide
        <input type="button" class="btn slide-one" value="Slide 1">
        <input type="button" class="btn slide-two" value="Slide 2">
        <input type="button" class="btn slide-three" value="Slide 3">
    </div>
</div>
<script>
    $(function(){
        // 初始化轮播
        $(".start-slide").click(function(){
            $("#myCarousel").carousel('cycle');
        });
        // 停止轮播
        $(".pause-slide").click(function(){
            $("#myCarousel").carousel('pause');
        });
        // 循环轮播到上一个项目
        $(".prev-slide").click(function(){
```

```
            $("#myCarousel").carousel('prev');
        });
        // 循环轮播到下一个项目
        $(".next-slide").click(function(){
            $("#myCarousel").carousel('next');
        });
        // 循环轮播到某个特定的帧
        $(".slide-one").click(function(){
            $("#myCarousel").carousel(0);
        });
        $(".slide-two").click(function(){
            $("#myCarousel").carousel(1);
        });
        $(".slide-three").click(function(){
            $("#myCarousel").carousel(2);
        });
    });
</script>

</body>
</html>
```

结果如下所示：



事件

下表列出了轮播（Carousel）插件中要用到的事件。这些事件可在函数中当钩子使用。

事件	描述	实例
slide.bs.carousel	当调用 slide 实例方法时立即触发该事件。	`$('#identifier').on('slide.bs.carousel', fu`
slid.bs.carousel	当轮播完成幻灯片过渡效果时触发该事件。	`$('#identifier').on('slid.bs.carousel', fun`

实例

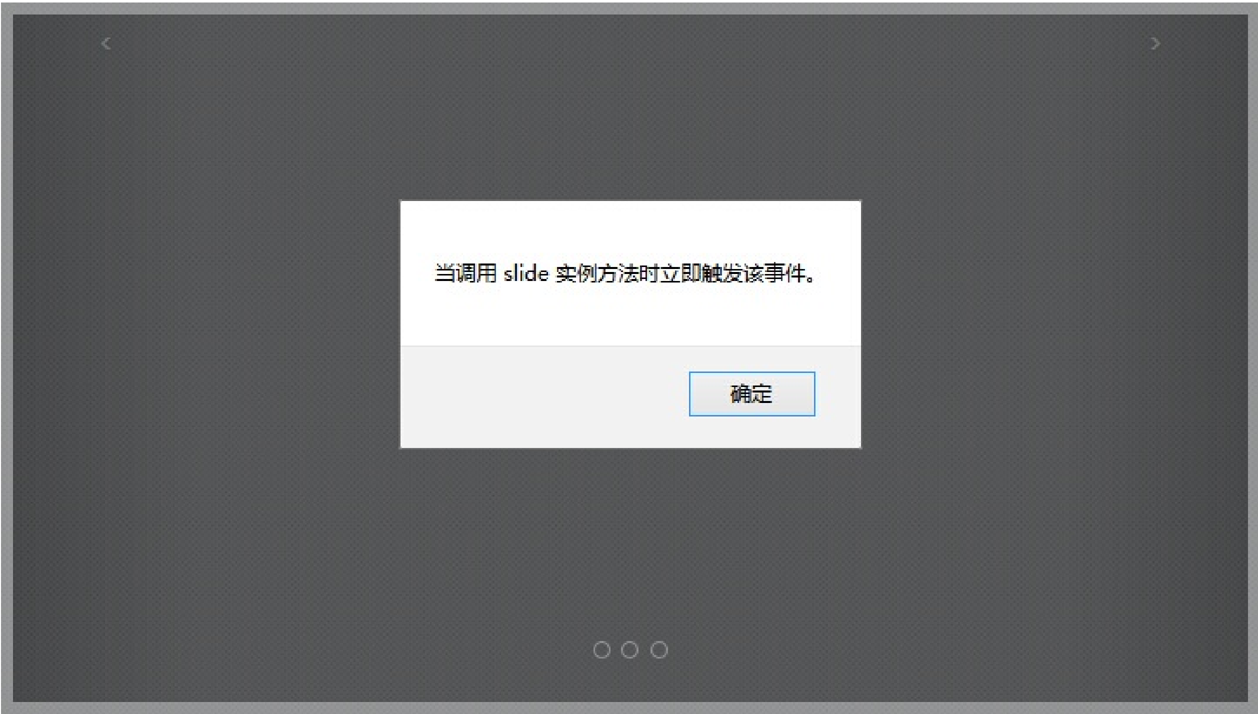
下面的实例演示了事件的用法：

```
<!DOCTYPE html>
<html>
<head>
   <title>Bootstrap 实例 - 轮播（Carousel）插件事件</title>
   <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
   <script src="/scripts/jquery.min.js"></script>
   <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div id="myCarousel" class="carousel slide">
   <!-- 轮播（Carousel）指标 -->
   <ol class="carousel-indicators">
      <li data-target="#myCarousel" data-slide-to="0"
         class="active"></li>
      <li data-target="#myCarousel" data-slide-to="1"></li>
      <li data-target="#myCarousel" data-slide-to="2"></li>
   </ol>
   <!-- 轮播（Carousel）项目 -->
   <div class="carousel-inner">
      <div class="item active">
         <img src="img/slide1.png" alt="First slide">
      </div>
      <div class="item">
         <img src="img/slide2.png" alt="Second slide">
      </div>
      <div class="item">
         <img src="img/slide3.png" alt="Third slide">
      </div>
   </div>
   <!-- 轮播（Carousel）导航 -->
   <a class="carousel-control left" href="#myCarousel"
      data-slide="prev">&lsaquo;</a>
   <a class="carousel-control right" href="#myCarousel"
      data-slide="next">&rsaquo;</a>
</div>
<script>
   $(function(){
      $('#myCarousel').on('slide.bs.carousel', function () {
         alert("当调用 slide 实例方法时立即触发该事件。");
      });
   });
</script>

</body>
</html>
```

结果如下所示：


当调用 slide 实例方法时立即触发该事件。
确定

Bootstrap 附加导航（Affix）插件

附加导航（Affix）插件允许某个 <div> 固定在页面的某个位置。您也可以在打开或关闭使用该插件之间进行切换。一个常见的例子是社交图标。它们将在某个位置开始，但当页面点击某个标记，该 <div> 会锁定在某个位置，不会随着页面其他部分一起滚动。

> 如果您想要单独引用该插件的功能，那么您需要引用 **affix.js**。或者，正如 Bootstrap 插件概览 一章中所提到，您可以引用 *bootstrap.js* 或压缩版的 *bootstrap.min.js*。

用法

您可以通过 data 属性或者通过 JavaScript 来使用附加导航（Affix）插件。

- 通过 **data** 属性：如需向元素添加附加导航（Affix）行为，只需要向需要监听的元素添加 **data-spy="affix"** 即可。请使用偏移来定义何时切换元素的锁定和移动。

实例

下面的实例演示了通过 data 属性使用附加导航（Affix）插件的用法：

```
&lt;!DOCTYPE html&gt;
&lt;html&gt;
&lt;head&gt;
&lt;meta charset="UTF-8"&gt;
&lt;title&gt;Bootstrap 附加导航（Affix）插件&lt;/title&gt;
&lt;link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet"&g
&lt;script src="/scripts/jquery.min.js"&gt;&lt;/script&gt;
&lt;script src="/bootstrap/js/bootstrap.min.js"&gt;&lt;/script&gt;

&lt;style type="text/css"&gt;
    /* Custom Styles */
    ul.nav-tabs{
        width: 140px;
        margin-top: 20px;
        border-radius: 4px;
        border: 1px solid #ddd;
        box-shadow: 0 1px 4px rgba(0, 0, 0, 0.067);
    }
    ul.nav-tabs li{
        margin: 0;
        border-top: 1px solid #ddd;
    }
    ul.nav-tabs li:first-child{
```

```
            border-top: none;
        }
        ul.nav-tabs li a{
            margin: 0;
            padding: 8px 16px;
            border-radius: 0;
        }
        ul.nav-tabs li.active a, ul.nav-tabs li.active a:hover{
            color: #fff;
            background: #0088cc;
            border: 1px solid #0088cc;
        }
        ul.nav-tabs li:first-child a{
            border-radius: 4px 4px 0 0;
        }
        ul.nav-tabs li:last-child a{
            border-radius: 0 0 4px 4px;
        }
        ul.nav-tabs.affix{
            top: 30px; /* Set the top position of pinned element */
        }
    &lt;/style&gt;
    &lt;/head&gt;
    &lt;body data-spy="scroll" data-target="#myScrollspy"&gt;
    &lt;div class="container"&gt;
        &lt;div class="jumbotron"&gt;
            &lt;h1&gt;Bootstrap Affix&lt;/h1&gt;
        &lt;/div&gt;
        &lt;div class="row"&gt;
            &lt;div class="col-xs-3" id="myScrollspy"&gt;
                &lt;ul class="nav nav-tabs nav-stacked" data-spy="affix
                    &lt;li class="active"&gt;&lt;a href="#section-1"&gt
                    &lt;li&gt;&lt;a href="#section-2"&gt;第二部分&lt;/a&
                    &lt;li&gt;&lt;a href="#section-3"&gt;第三部分&lt;/a&
                    &lt;li&gt;&lt;a href="#section-4"&gt;第四部分&lt;/a&
                    &lt;li&gt;&lt;a href="#section-5"&gt;第五部分&lt;/a&
                &lt;/ul&gt;
            &lt;/div&gt;
            &lt;div class="col-xs-9"&gt;
                &lt;h2 id="section-1"&gt;第一部分&lt;/h2&gt;
                &lt;p&gt;Lorem ipsum dolor sit amet, consectetur adipis
                &lt;p&gt;Vestibulum quis quam ut magna consequat faucib
                &lt;hr&gt;
                &lt;h2 id="section-2"&gt;第二部分&lt;/h2&gt;
                &lt;p&gt;Nullam hendrerit justo non leo aliquet imperdi
                &lt;p&gt;Vestibulum consectetur scelerisque lacus, ac t
                &lt;hr&gt;
                &lt;h2 id="section-3"&gt;第三部分&lt;/h2&gt;
                &lt;p&gt;Integer pulvinar leo id risus pellentesque ves
                &lt;p&gt;Phasellus vitae suscipit justo. Mauris pharetr
                &lt;p&gt;Quisque pharetra velit id velit iaculis pretiu
                &lt;hr&gt;
                &lt;h2 id="section-4"&gt;第四部分&lt;/h2&gt;
```

```
            &lt;p&gt;Suspendisse a orci facilisis, dignissim tortor
            &lt;p&gt;Vestibulum quis quam ut magna consequat faucik
            &lt;p&gt;Phasellus fermentum, neque sit amet sodales te
            &lt;p&gt;Ut ut risus nisl. Fusce porttitor eros at magr
            &lt;hr&gt;
            &lt;h2 id="section-5"&gt;第五部分&lt;/h2&gt;
            &lt;p&gt;Nam eget purus nec est consectetur vehicula. N
            &lt;p&gt;Vivamus mattis accumsan erat, vel convallis ri
            &lt;p&gt;Suspendisse a orci facilisis, dignissim tortor
            &lt;p&gt;Morbi sed fermentum ipsum. Morbi a orci vulput
            &lt;p&gt;Sed vitae lobortis diam, id molestie magna. Al
        &lt;/div&gt;
    &lt;/div&gt;
&lt;/div&gt;
&lt;/body&gt;
&lt;/html&gt;
```

结果如下所示：



- 通过 **JavaScript**：您可以通过 JavaScript 手动为某个元素添加附加导航（Affix），如下所示：

```
$('#myAffix').affix({
   offset: {
      top: 100, bottom: function () {
         return (this.bottom =
            $('.bs-footer').outerHeight(true))
         }
      }
})
```

实例

下面的实例演示了通过 JavaScript 使用附加导航（Affix）插件的用法：

```
&lt;!DOCTYPE html&gt;
&lt;html&gt;
&lt;head&gt;
&lt;meta charset="UTF-8"&gt;
&lt;title&gt;Bootstrap 附加导航（Affix）插件&lt;/title&gt;
&lt;link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet"&g
&lt;script src="/scripts/jquery.min.js"&gt;&lt;/script&gt;
&lt;script src="/bootstrap/js/bootstrap.min.js"&gt;&lt;/script&gt;
&lt;style type="text/css"&gt;
    /* Custom Styles */
    ul.nav-tabs{
        width: 140px;
        margin-top: 20px;
        border-radius: 4px;
        border: 1px solid #ddd;
        box-shadow: 0 1px 4px rgba(0, 0, 0, 0.067);
    }
    ul.nav-tabs li{
        margin: 0;
        border-top: 1px solid #ddd;
    }
    ul.nav-tabs li:first-child{
        border-top: none;
    }
    ul.nav-tabs li a{
        margin: 0;
        padding: 8px 16px;
        border-radius: 0;
    }
    ul.nav-tabs li.active a, ul.nav-tabs li.active a:hover{
        color: #fff;
        background: #0088cc;
        border: 1px solid #0088cc;
    }
    ul.nav-tabs li:first-child a{
        border-radius: 4px 4px 0 0;
    }
```

```
    ul.nav-tabs li:last-child a{
        border-radius: 0 0 4px 4px;
    }
    ul.nav-tabs.affix{
        top: 30px; /* Set the top position of pinned element */
    }
&lt;/style&gt;
&lt;script type="text/javascript"&gt;
$(document).ready(function(){
    $("#myNav").affix({
        offset: {
            top: 125
         }
    });
});
&lt;/script&gt;
&lt;/head&gt;
&lt;body data-spy="scroll" data-target="#myScrollspy"&gt;
&lt;div class="container"&gt;
    &lt;div class="jumbotron"&gt;
        &lt;h1&gt;Bootstrap Affix&lt;/h1&gt;
    &lt;/div&gt;
    &lt;div class="row"&gt;
        &lt;div class="col-xs-3" id="myScrollspy"&gt;
            &lt;ul class="nav nav-tabs nav-stacked" id="myNav"&gt;
                &lt;li class="active"&gt;&lt;a href="#section-1"&g
                &lt;li&gt;&lt;a href="#section-2"&gt;第二部分&lt;/a&
                &lt;li&gt;&lt;a href="#section-3"&gt;第三部分&lt;/a&
                &lt;li&gt;&lt;a href="#section-4"&gt;第四部分&lt;/a&
                &lt;li&gt;&lt;a href="#section-5"&gt;第五部分&lt;/a&
            &lt;/ul&gt;
        &lt;/div&gt;
        &lt;div class="col-xs-9"&gt;
            &lt;h2 id="section-1"&gt;第一部分&lt;/h2&gt;
            &lt;p&gt;Lorem ipsum dolor sit amet, consectetur adipis
            &lt;p&gt;Vestibulum quis quam ut magna consequat faucib
            &lt;hr&gt;
            &lt;h2 id="section-2"&gt;第二部分&lt;/h2&gt;
            &lt;p&gt;Nullam hendrerit justo non leo aliquet imperdi
            &lt;p&gt;Vestibulum consectetur scelerisque lacus, ac f
            &lt;hr&gt;
            &lt;h2 id="section-3"&gt;第三部分&lt;/h2&gt;
            &lt;p&gt;Integer pulvinar leo id risus pellentesque ves
            &lt;p&gt;Phasellus vitae suscipit justo. Mauris pharetr
            &lt;p&gt;Quisque pharetra velit id velit iaculis pretiu
            &lt;hr&gt;
            &lt;h2 id="section-4"&gt;第四部分&lt;/h2&gt;
            &lt;p&gt;Suspendisse a orci facilisis, dignissim tortor
            &lt;p&gt;Vestibulum quis quam ut magna consequat faucib
            &lt;p&gt;Phasellus fermentum, neque sit amet sodales te
            &lt;p&gt;Ut ut risus nisl. Fusce porttitor eros at magn
            &lt;hr&gt;
            &lt;h2 id="section-5"&gt;第五部分&lt;/h2&gt;
```

```
            &lt;p&gt;Nam eget purus nec est consectetur vehicula. N
            &lt;p&gt;Vivamus mattis accumsan erat, vel convallis ri
            &lt;p&gt;Suspendisse a orci facilisis, dignissim tortor
            &lt;p&gt;Morbi sed fermentum ipsum. Morbi a orci vulput
            &lt;p&gt;Sed vitae lobortis diam, id molestie magna. Al
        &lt;/div&gt;
    &lt;/div&gt;
&lt;/div&gt;
&lt;/body&gt;
&lt;/html&gt;
```

结果如下所示：



通过 CSS 定位

在上面两种使用附加导航（Affix）插件的方式中，您都必须通过 CSS 定位内容。附加导航（Affix）插件在三种 class 之间切换，每种 class 都呈现了特定的状态：*.affix*、*.affix-top* 和 *.affix-bottom*。请按照下面的步骤，来为这三种状态设置您自己的 CSS（不依赖此插件）。

- 在开始时，插件添加 **.affix-top** 来指示元素在它的最顶端位置。这个时候不需要任何的 CSS 定位。
- 当滚动经过添加了附加导航（Affix）的元素时，应触发实际的附加导航（Affix）。此时 **.affix** 会替代 **.affix-top**，同时设置 **position: fixed;**（由 Bootstrap 的 CSS 代码提供）。
- 如果定义了底部偏移，当滚动到达该位置时，应把 **.affix** 替换为 **.affix-bottom**。由于偏移是可选的，假如设置了该偏移，则要求同时设置适当的 CSS。在这种情况下，请在必要的时候添加 **position: absolute;**。

选项

有一些选项是通过 data 属性或 JavaScript 来传递的。下表列出了这些选项：

选项名称	类型/默认值	**Data** 属性名称	描述
offset	number | function | object 默认值：*10*	data-offset	当计算滚动位置时，距离顶部的偏移像素。如果设置了一个数字，则该偏移量的值将被应用在顶部和底部。如果设置了一个对象偏移，则其值形如 offset: { top: 10 } 或 offset: { top: 10, bottom: 5 }。如果需要动态计算偏移，请使用函数。

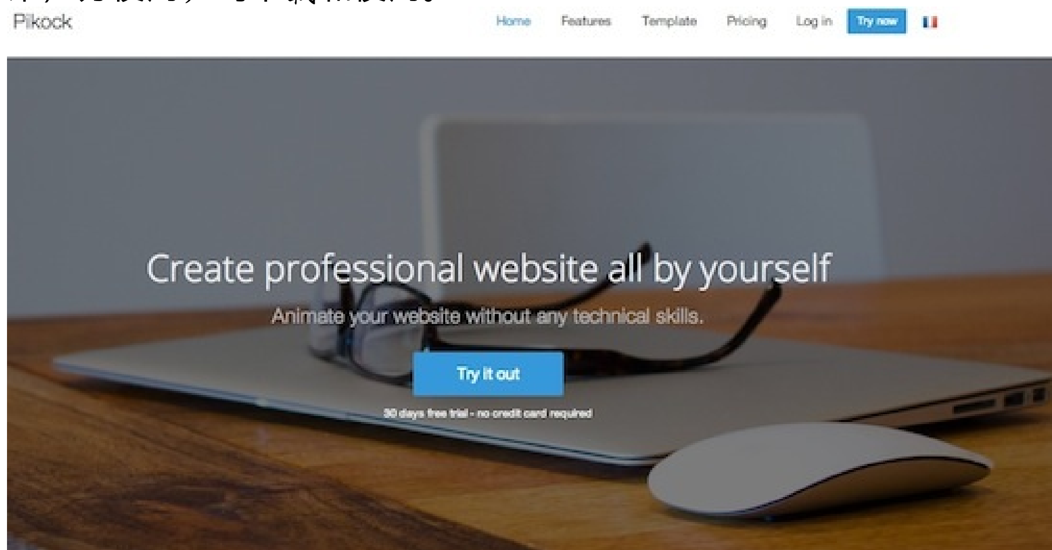
Bootstrap 其它

Bootstrap UI 编辑器

以下是 15 款最好的 Bootstrap 编辑器或者是在线编辑工具。

1. Bootstrap Magic

这是一个 Bootstrap 主题生成器，使用最新的 Bootstrap 3 版本和 Angular JS 版本，提供一个鲜活的用户修改预览。它包括了各种各样的导入，一个颜色选择器和智能的预先输入。更神奇的是，Bootstrap 会根据用户的每个选择来重新简历框架，方便用户的下载和使用。



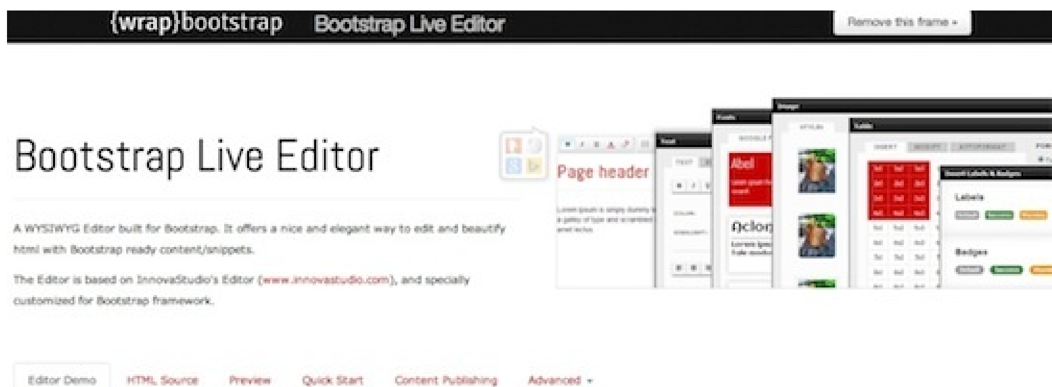
2. BootSwatchr

BootSwatchr 是由 Drew Strickiand 独立开发和维护的，是唯一支持从右到左语言显示的 Bootstrap 自定义构建工具，这也是它的特色之一。BootSwatchr 是个可视化的工具，可以从底层开始创建 Bootstrap 主题。



3. Bootstrap Live Editor

Bootstrap Live Editor 是个所见即所得的构建 Bootstrap 的编辑器。在 Bootstrap 准备好的内容和代码片段的基础上提供一个又好又优雅的方法来编辑和美化 html，也因此它是个明智的选择。除此之外，它还包括高级的选项比如按钮配置，自定义标签和为用户做最明智最有益的选择。



4. Fancy Boot

Fancyboot 是个非常有用的 Bootstrap 自定义工具，允许用户编辑 Bootstrap 的配置，自动预览他们的修改结果。通过滑动菜单可以选择需要使用的插件和组件。当用户做好了相应的修改，可以下载一个包含用户所使用的 jQuery 插件的编译好的和简化后的 CSS 文件。



5. Style Bootstrap

Style Bootstrap 是个非常棒的内置浏览器的 GUI 实用工具，帮助用户自定义 Twitter Bootstrap 外观。它非常容易进行自定义操作。从 body 风格，字体排版，按钮，导航，格式，通知提示和其他用户很容易根据自己的意愿自定义的元素。除了强大的自定义功能，它还可以很容易生成可下载的 CSS 文件。

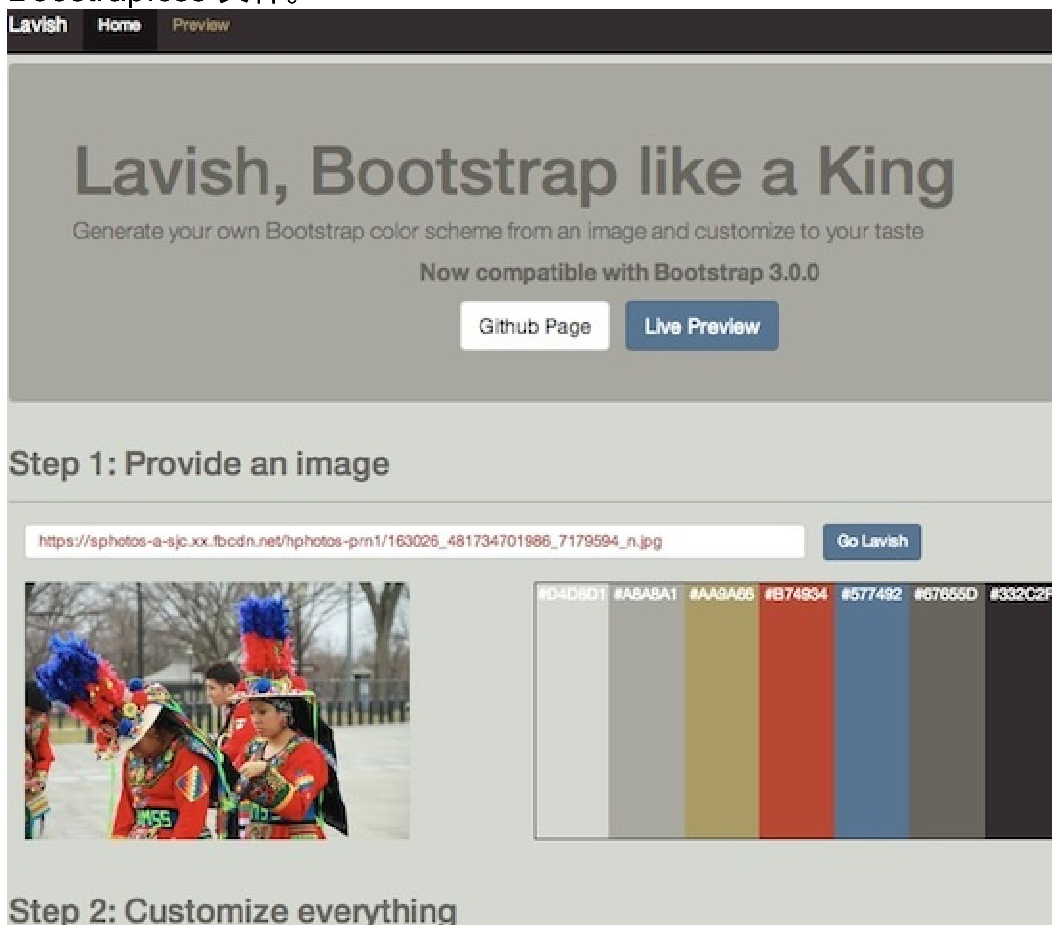
Style Bootstrap 是个非常棒的内置浏览器的 GUI 实用工具，帮助用户自定义 Twitter Bootstrap 外观。它非常容易进行自定义操作。从 body 风格，字体排版，按钮，导航，格式，通知提示和其他用户很容易根据自己的意愿自定义的元素。除了强大的自定义功能，它还可以很容易生成可下载的 CSS 文件。

Buttons



6. Lavish

Lavish 是个帮助用户从一个图像中定制自己喜欢的 Bootstrap 配色方案。用户可以从图像中选择各种颜色用在 body 背景，链接，头部，菜单，标签，文本，下拉菜单等等元素上。一旦你确定了 Bootstrap 配色方案，就可以很方便的下载相应的 Boostrap.css 文件。



Bootstrap ThemeRoller 是个直观的 web 应用程序， 提供一个可视化的界面让用户按照自己的风格创建属于自己的 Bootstrap 主题。有了这个工具，用户可以自定义颜色，大小，字体，格式，表，按钮，警告信息，导航条，菜单等等元素。 这个应用还会帮用户创建的每种风格都生成唯一的一个 URL，这样就可以保存用户所有的设置。更不可思议的是，当用户离开的时候，用户可以停止或者恢复自己定制的



页面。

8. LayoutIt!

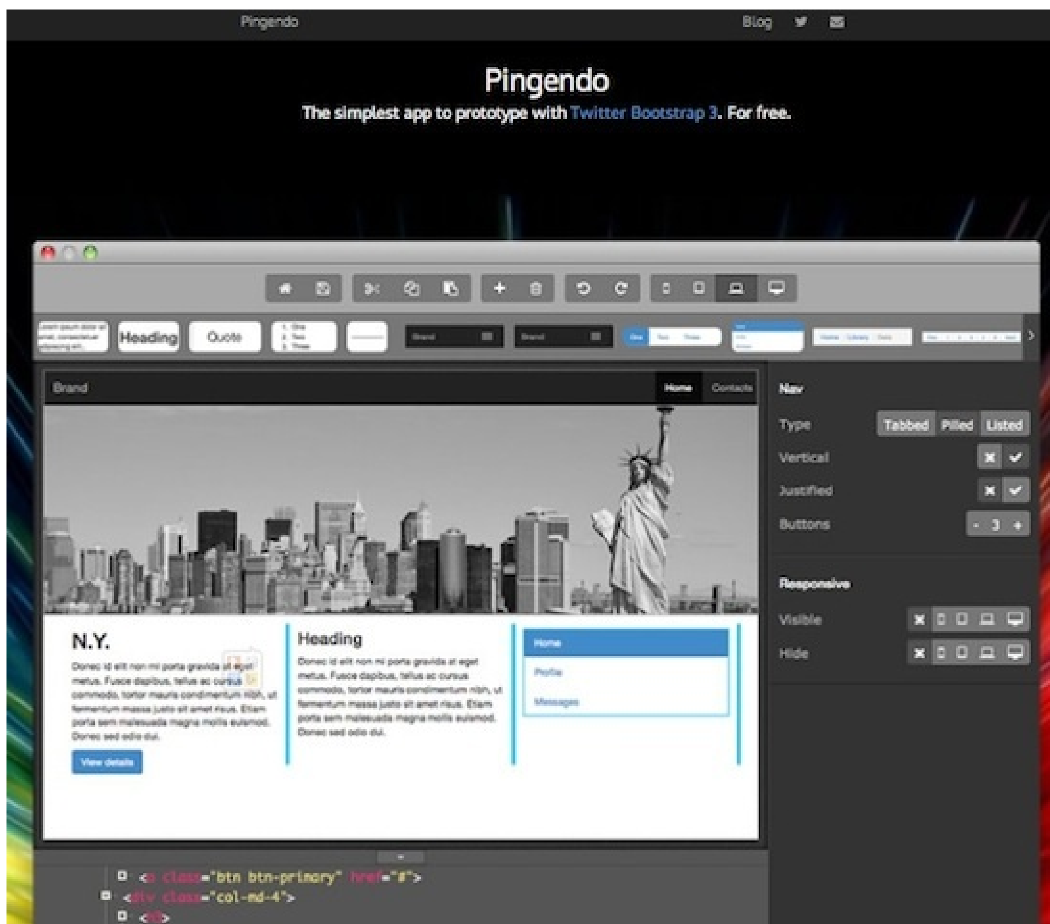
LayoutIt！拥有拖拽接口的功能，能简单迅速的构建一个 Bootstrap 前端代码。LayoutIt！ 兼容任何的编程语言，允许用户下载 HTML，在这里自由的进行编码设



计。

9. Pingendo

Pingendo 是个可视化的桌面应用程序，允许用户给予流行的 Twitter Bootstrap 框架创建响应式 web 页面原型。Pingendo 还提供大量的现成的布局，拖拽元素的位置，调整大小和自定义的功能。用户可以自行优化他们的页面，插入他们自己的内

容。

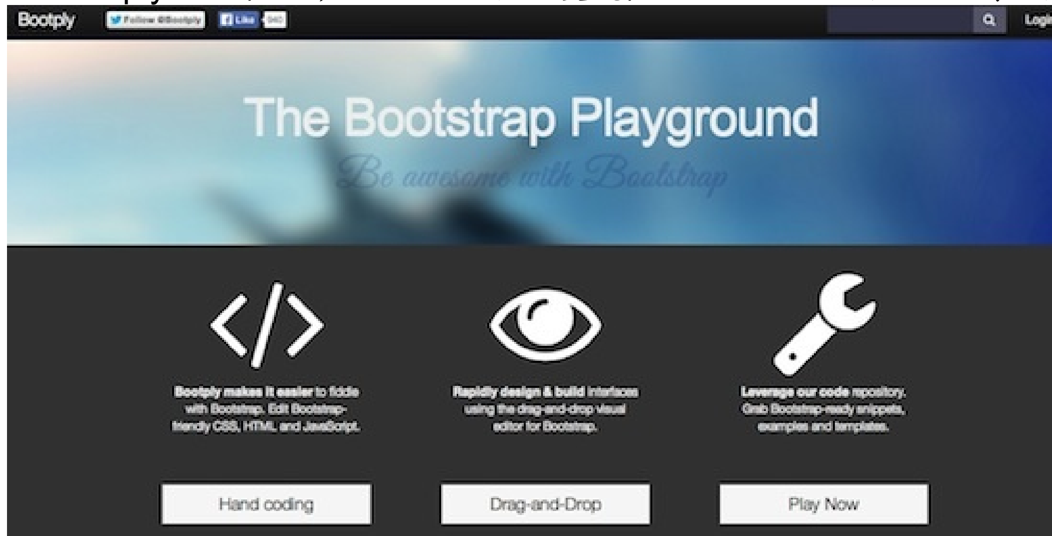
10. Kickstrap

Kickstrap 是个 Bootstrap UI 编辑器，使用 Firebase 作为一个后台服务。它无缝的接合了 Bootstrap 和顶级的 web 技术。更厉害的是，它可以运行一个已验证的，不需要本地后台支持，数据库驱动的 web 应用。



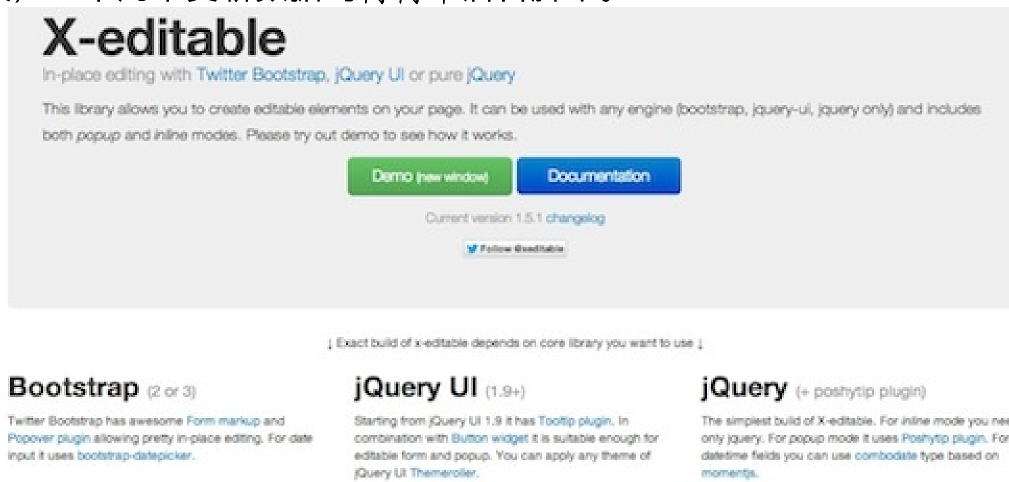
11. Bootply

Bootply 是一款很棒的 Bootstrap UI 编辑工具，尽可能简单的让用户编辑 Bootstrap 友好的 CSS，HTML 和 Javascript。同时也是一个可拖拽的可视化编辑器，用来设计，创建原型和测试 Bootstrap 框架。Bootply 有个代码存储库，里面包括了 Bootstrap 代码片段，示例和模板。除了以上这些，更吸引人的是它有个非常活跃的 Bootply 社区，用户可以很方便的发现，分享和展示相关的代码和片段。



12. X-editable

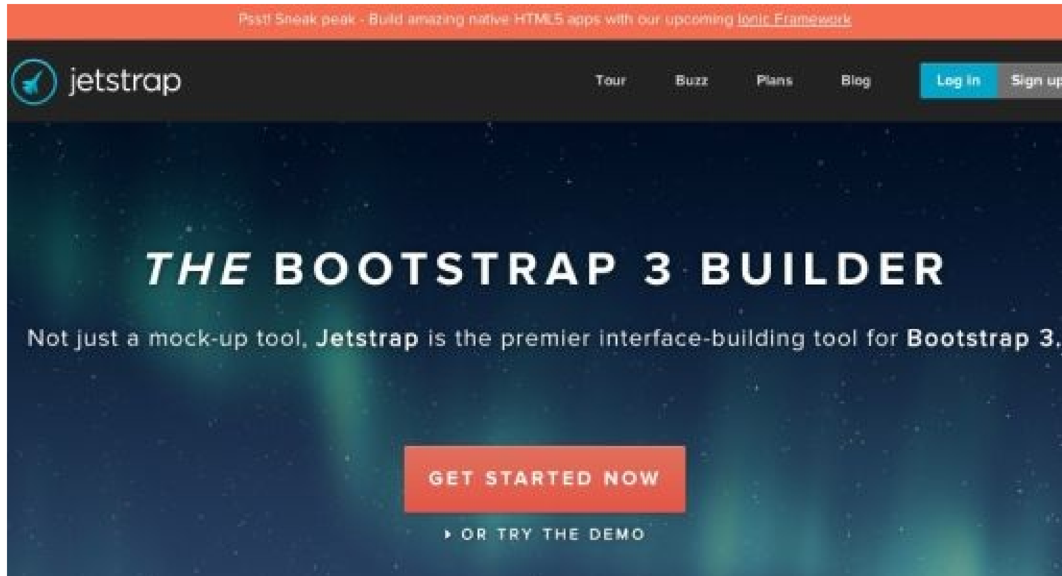
X-Editable 是一个扩展库，允许用户创建可编辑的元素，比如页面弹出和内联模式。她提供本地的 Bootstrap，jQuery UI 或者是纯 jQuery 编辑功能。此 Bootstrap UI 编辑器拥有许多很强大的功能，包括客户端，服务端验证，支持键盘，live 事件，可定制的容器空间等等。总而言之，它提供非常快速的开发功能，易于删除字段，一个用来更新数据的简简单后台脚本。



13. Jetstrap

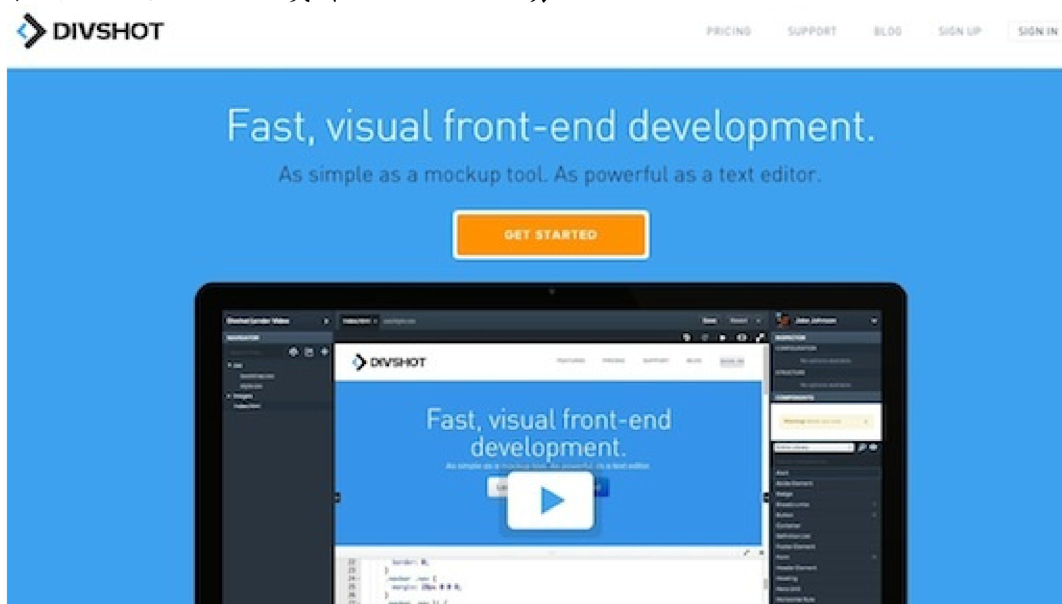
不要错误的以为 Jetstrap 只是个简单的实体模型工具，它是 100% 基于 web 接口的 Twitter Bootstrap 构建工具。用户不需要下载任何类型的软件就可以使用 Jetstrap。用户只需要登录，然后构建自己的项目，就可以随时随地进行访问了。

构建功能主要是提供给开发者和设计者的，用户很容易就可以通过 Jetstrap 生成漂亮的网站，而且运行速度非常快



14. DivShot

DivShot 是个快速的，可视化的前端开发文本编辑器，跟 mockup 工具一样的简单整洁。包括各种框架需要的各种定制的组件库，这是它提供的最好的工具之一。除此之外还包括了一个主题转换器，允许用户修改主题。试验版本是免费的，过了试验期就需要自己付费维持原来的服务了。



15. PaintStrap

PaintStrap 使用 Adobe kuler/COLOURlovers 颜色组合来生成漂亮的 Twitter Bootstrap 主题。这个工具特别容易安装，同时很容易下载 CSS 文件并且也很容易替换成另一个标准的文件。


PaintStrap
Home
Gallery
English
日本語
PaintStrap
Generate beautiful Twitter Bootstrap themes using the Adobe kuler / COLOURlovers color scheme
Step 1: Input a color scheme
Step 2: Set colors
Step 3: Download CSS files
Step 1: Input a color scheme
Use an Adobe kuler color scheme(Color Theme)
Please input a theme ID or a permalink URL:
E.g. 1084168, 1006374, 2043016
Use a COLOURlovers color scheme(Palette)
Please input a palette ID or a permalink URL:
E.g. 694737, 678929, 77121

Bootstrap HTML编码规范

语法

- 用两个空格来代替制表符（tab） -- 这是唯一能保证在所有环境下获得一致展现的方法。
- 嵌套元素应当缩进一次（即两个空格）。
- 对于属性的定义，确保全部使用双引号，绝不要使用单引号。
- 不要在自闭合（self-closing）元素的尾部添加斜线 -- HTML5 规范中明确说明这是可选的。
- 不要省略可选的结束标签（closing tag）（例如，`</li>` 或 `</body>` ）。

实例：

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page title</title>
  </head>
  <body>
    <img src="images/company-logo.png" alt="Company">
    <h1 class="hello-world">Hello, world!</h1>
  </body>
</html>
```

HTML5 doctype

为每个 HTML 页面的第一行添加标准模式（standard mode）的声明，这样能够确保在每个浏览器中拥有一致的展现。

实例：

```
<!DOCTYPE html>
<html>
  <head>
  </head>
</html>
```

语言属性

根据 HTML5 规范：

强烈建议为 html 根元素指定 lang 属性，从而为文档设置正确的语言。这将有助于语音合成工具确定其所应该采用的发音，有助于翻译工具确定其翻译时所应遵守的规则等等。

更多关于 `lang` 属性的知识可以从 此规范 中了解。

这里列出了语言代码表。

```
<html lang="zh-CN">
  <!-- ... -->
</html>
```

IE 兼容模式

IE 支持通过特定的 `<meta>` 标签来确定绘制当前页面所应该采用的 IE 版本。除非有强烈的特殊需求，否则最好是设置为 **edge mode**，从而通知 IE 采用其所支持的最新的模式。

```
<meta http-equiv="X-UA-Compatible" content="IE=Edge">
```

字符编码

通过明确声明字符编码，能够确保浏览器快速并容易的判断页面内容的渲染方式。这样做的好处是，可以避免在 HTML 中使用字符实体标记（character entity），从而全部与文档编码一致（一般采用 UTF-8 编码）。

```
<head>
  <meta charset="UTF-8">
</head>
```

引入 CSS 和 JavaScript 文件

根据 HTML5 规范，在引入 CSS 和 JavaScript 文件时一般不需要指定 `type` 属性，因为 `text/css` 和 `text/javascript` 分别是它们的默认值。

HTML5 spec links

- Using link
- Using style
- Using script

```
<!-- External CSS -->
<link rel="stylesheet" href="code-guide.css">

<!-- In-document CSS -->
<style>
  /* ... */
</style>

<!-- JavaScript -->
<script src="code-guide.js"></script>
```

实用为王

尽量遵循 HTML 标准和语义，但是不要以牺牲实用性为代价。任何时候都要尽量使用最少的标签并保持最小的复杂度。

属性顺序

HTML 属性应当按照以下给出的顺序依次排列，确保代码的易读性。

- `class`
- `id` , `name`
- `data-*`
- `src` , `for` , `type` , `href`
- `title` , `alt`
- `aria-*` , `role`

class 用于标识高度可复用组件，因此应该排在首位。id 用于标识具体组件，应当谨慎使用（例如，页面内的书签），因此排在第二位。

```
<a class="..." id="..." data-modal="toggle" href="#">
  Example link
</a>

<input class="form-control" type="text">

<img src="..." alt="...">
```

布尔（boolean）型属性

布尔型属性可以在声明时不赋值。XHTML 规范要求为其赋值，但是 HTML5 规范不需要。

更多信息请参考 WhatWG section on boolean attributes：

元素的布尔型属性如果有值，就是 true，如果没有值，就是 false。

如果一定要为其赋值的话，请参考 WhatWG 规范：

如果属性存在，其值必须是空字符串或 [...] 属性的规范名称，并且不要再收尾添加空白符。

简单来说，就是不用赋值。

```
<input type="text" disabled>

<input type="checkbox" value="1" checked>

<select>
  <option value="1" selected>1</option>
</select>
```

减少标签的数量

编写 HTML 代码时，尽量避免多余的父元素。很多时候，这需要迭代和重构来实现。请看下面的案例：

```
<!-- Not so great -->
<span class="avatar">
  <img src="...">
</span>

<!-- Better -->
<img class="avatar" src="...">
```

JavaScript 生成的标签

通过 JavaScript 生成的标签让内容变得不易查找、编辑，并且降低性能。能避免时尽量避免。

Bootstrap CSS编码规范

语法

- 用两个空格来代替制表符（tab） -- 这是唯一能保证在所有环境下获得一致展现的方法。
- 为选择器分组时，将单独的选择器单独放在一行。
- 为了代码的易读性，在每个声明块的左花括号前添加一个空格。
- 声明块的右花括号应当单独成行。
- 每条声明语句的 `:` 后应该插入一个空格。
- 为了获得更准确的错误报告，每条声明都应该独占一行。
- 所有声明语句都应当以分号结尾。最后一条声明语句后面的分号是可选的，但是，如果省略这个分号，你的代码可能更易出错。
- 对于以逗号分隔的属性值，每个逗号后面都应该插入一个空格（例如，`box-shadow` ）。
- 不要在 `rgb()` 、 `rgba()` 、 `hsl()` 、 `hsla()` 或 `rect()` 值的内部的逗号后面插入空格。这样利于从多个属性值（既加逗号也加空格）中区分多个颜色值（只加逗号，不加空格）。
- 对于属性值或颜色参数，省略小于 1 的小数前面的 0 （例如， `.5` 代替 `0.5` ； `-.5px` 代替 `-0.5px` ）。
- 十六进制值应该全部小写，例如， `#fff` 。在扫描文档时，小写字符易于分辨，因为他们的形式更易于区分。
- 尽量使用简写形式的十六进制值，例如，用 `#fff` 代替 `#ffffff` 。
- 为选择器中的属性添加双引号，例如， `input[type="text"]` 。只有在某些情况下是可选的，但是，为了代码的一致性，建议都加上双引号。
- 避免为 0 值指定单位，例如，用 `margin: 0;` 代替 `margin: 0px;` 。

对于这里用到的术语有疑问吗？请参考 Wikipedia 上的 syntax section of the Cascading Style Sheets article。

```
/* Bad CSS */
.selector, .selector-secondary, .selector[type=text] {
  padding:15px;
  margin:0px 0px 15px;
  background-color:rgba(0, 0, 0, 0.5);
  box-shadow:0px 1px 2px #CCC,inset 0 1px 0 #FFFFFF
}

/* Good CSS */
.selector,
.selector-secondary,
.selector[type="text"] {
  padding: 15px;
  margin-bottom: 15px;
  background-color: rgba(0,0,0,.5);
  box-shadow: 0 1px 2px #ccc, inset 0 1px 0 #fff;
}
```

声明顺序

相关的属性声明应当归为一组，并按照下面的顺序排列：

1. Positioning
2. Box model
3. Typographic
4. Visual

由于定位（positioning）可以从正常的文档流中移除元素，并且还能覆盖盒模型（box model）相关的样式，因此排在首位。盒模型排在第二位，因为它决定了组件的尺寸和位置。

其他属性只是影响组件的内部（*inside*）或者是不影响前两组属性，因此排在后面。

完整的属性列表及其排列顺序请参考 Recess。

```
.declaration-order {
  /* Positioning */
  position: absolute;
  top: 0;
  right: 0;
  bottom: 0;
  left: 0;
  z-index: 100;

  /* Box-model */
  display: block;
  float: right;
  width: 100px;
  height: 100px;

  /* Typography */
  font: normal 13px "Helvetica Neue", sans-serif;
  line-height: 1.5;
  color: #333;
  text-align: center;

  /* Visual */
  background-color: #f5f5f5;
  border: 1px solid #e5e5e5;
  border-radius: 3px;

  /* Misc */
  opacity: 1;
}
```

不要使用 `@import`

与 `<link>` 标签相比，`@import` 指令要慢很多，不光增加了额外的请求次数，还会导致不可预料的问题。替代办法有以下几种：

- 使用多个 `<link>` 元素
- 通过 Sass 或 Less 类似的 CSS 预处理器将多个 CSS 文件编译为一个文件
- 通过 Rails、Jekyll 或其他系统中提供过 CSS 文件合并功能

请参考 Steve Souders 的文章了解更多知识。

```
<!-- Use link elements -->
<link rel="stylesheet" href="core.css">

<!-- Avoid @imports -->
<style>
  @import url("more.css");
</style>
```

媒体查询（Media query）的位置

将媒体查询放在尽可能相关规则的附近。不要将他们打包放在一个单一样式文件中或者放在文档底部。如果你把他们分开了，将来只会被大家遗忘。下面给出一个典型的实例。

```
.element { ... }
.element-avatar { ... }
.element-selected { ... }

@media (min-width: 480px) {
  .element { ...}
  .element-avatar { ... }
  .element-selected { ... }
}
```

带前缀的属性

当使用特定厂商的带有前缀的属性时，通过缩进的方式，让每个属性的值在垂直方向对齐，这样便于多行编辑。

在 Textmate 中，使用 **Text → Edit Each Line in Selection** (??A)。在 Sublime Text 2 中，使用 **Selection → Add Previous Line** (??↑) 和 **Selection → Add Next Line** (??↓)。

```
/* Prefixed properties */
.selector {
  -webkit-box-shadow: 0 1px 2px rgba(0,0,0,.15);
          box-shadow: 0 1px 2px rgba(0,0,0,.15);
}
```

单行规则声明

对于只包含一条声明的样式，为了易读性和便于快速编辑，建议将语句放在同一行。对于带有多条声明的样式，还是应当将声明分为多行。

这样做的关键因素是为了错误检测 -- 例如，CSS 校验器指出在 183 行有语法错误。如果是单行单条声明，你就不会忽略这个错误；如果是单行多条声明的话，你就要仔细分析避免漏掉错误了。

```
/* Single declarations on one line */
.span1 { width: 60px; }
.span2 { width: 140px; }
.span3 { width: 220px; }

/* Multiple declarations, one per line */
.sprite {
  display: inline-block;
  width: 16px;
  height: 15px;
  background-image: url(../img/sprite.png);
}
.icon           { background-position: 0 0; }
.icon-home      { background-position: 0 -20px; }
.icon-account   { background-position: 0 -40px; }
```

简写形式的属性声明

在需要显示地设置所有值的情况下，应当尽量限制使用简写形式的属性声明。常见的滥用简写属性声明的情况如下：

- `padding`
- `margin`
- `font`
- `background`
- `border`
- `border-radius`

大部分情况下，我们不需要为简写形式的属性声明指定所有值。例如，HTML 的 heading 元素只需要设置上、下边距（margin）的值，因此，在必要的时候，只需覆盖这两个值就可以。过度使用简写形式的属性声明会导致代码混乱，并且会对属性值带来不必要的覆盖从而引起意外的副作用。

MDN（Mozilla Developer Network）上一片非常好的关于shorthand properties 的文章，对于不太熟悉简写属性声明及其行为的用户很有用。

```
/* Bad example */
.element {
  margin: 0 0 10px;
  background: red;
  background: url("image.jpg");
  border-radius: 3px 3px 0 0;
}

/* Good example */
.element {
  margin-bottom: 10px;
  background-color: red;
  background-image: url("image.jpg");
  border-top-left-radius: 3px;
  border-top-right-radius: 3px;
}
```

Less 和 Sass 中的嵌套

避免非必要的嵌套。这是因为虽然你可以使用嵌套，但是并不意味着应该使用嵌套。只有在必须将样式限制在父元素内（也就是后代选择器），并且存在多个需要嵌套的元素时才使用嵌套。

```
// Without nesting
.table > thead > tr > th { … }
.table > thead > tr > td { … }

// With nesting
.table > thead > tr {
  > th { … }
  > td { … }
}
```

注释

代码是由人编写并维护的。请确保你的代码能够自描述、注释良好并且易于他人理解。好的代码注释能够传达上下文关系和代码目的。不要简单地重申组件或 class 名称。

对于较长的注释，务必书写完整的句子；对于一般性注解，可以书写简洁的短语。

```
/* Bad example */
/* Modal header */
.modal-header {
  ...
}

/* Good example */
/* Wrapping element for .modal-title and .modal-close */
.modal-header {
  ...
}
```

class 命名

- class 名称中只能出现小写字符和破折号（dashe）（不是下划线，也不是驼峰命名法）。破折号应当用于相关 class 的命名（类似于命名空间）（例如，`.btn` 和 `.btn-danger`）。
- 避免过度任意的简写。`.btn` 代表 *button*，但是 `.s` 不能表达任何意思。
- class 名称应当尽可能短，并且意义明确。
- 使用有意义的名称。使用有组织的或目的明确的名称，不要使用表现形式（presentational）的名称。
- 基于最近的父 class 或基本（base） class 作为新 class 的前缀。
- 使用 `.js-*` class 来标识行为（与样式相对），并且不要将这些 class 包含到 CSS 文件中。

在为 Sass 和 Less 变量命名是也可以参考上面列出的各项规范。

```
/* Bad example */
.t { ... }
.red { ... }
.header { ... }

/* Good example */
.tweet { ... }
.important { ... }
.tweet-header { ... }
```

选择器

- 对于通用元素使用 class ，这样利于渲染性能的优化。
- 对于经常出现的组件，避免使用属性选择器（例如，`[class^="..."]`）。浏览器的性能会受到这些因素的影响。
- 选择器要尽可能短，并且尽量限制组成选择器的元素个数，建议不要超过 3 。
- 只有在必要的时候才将 class 限制在最近的父元素内（也就是后代选择器）（例如，不使用带前缀的 class 时 -- 前缀类似于命名空间）。

扩展阅读：

- Scope CSS classes with prefixes
- Stop the cascade

```
/* Bad example */
span { ... }
.page-container #stream .stream-item .tweet .tweet-header .username
.avatar { ... }

/* Good example */
.avatar { ... }
.tweet-header .username { ... }
.tweet .avatar { ... }
```

代码组织

- 以组件为单位组织代码段。
- 制定一致的注释规范。
- 使用一致的空白符将代码分隔成块，这样利于扫描较大的文档。
- 如果使用了多个 CSS 文件，将其按照组件而非页面的形式分拆，因为页面会被重组，而组件只会被移动。

```
/*
 * Component section heading
 */

.element { ... }

/*
 * Component section heading
 *
 * Sometimes you need to include optional context for the entire co
 */

.element { ... }

/* Contextual sub-component or modifer */
.element-heading { ... }
```

编辑器配置

将你的编辑器按照下面的配置进行设置，以避免常见的代码不一致和差异：

- 用两个空格代替制表符（soft-tab 即用空格代表 tab 符）。

- 保存文件时，删除尾部的空白符。
- 设置文件编码为 UTF-8。
- 在文件结尾添加一个空白行。

参照文档并将这些配置信息添加到项目的 `.editorconfig` 文件中。例如：Bootstrap 中的 .editorconfig 实例。更多信息请参考 about EditorConfig。

免责声明

W3School提供的内容仅用于培训。我们不保证内容的正确性。通过使用本站内容随之而来的风险与本站无关。W3School简体中文版的所有内容仅供测试，对任何法律问题及风险不承担任何责任。